

CS C121

Probabilistic Models in Computational Genomics

Probability, genomic data, sequence models, latent variables, inference, and responsible interpretation

How these notes are written. I wrote each chapter the way I wish the material had been explained the first time: the real idea, the point where I usually get stuck, the simplest mental model that still stays honest, and a worked decision instead of a polished answer appearing from nowhere. Every chapter closes with practice and a fully written solution. If a sentence sounds impressive but does not help me solve or explain something, it does not belong here.

Daniel Mastick

Curriculum map, working notes, practice, and solutions

Course map

Week	Core thread	What should be solid by the end
1	Genomic data as a probabilistic object	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
2	Conditional probability and Bayesian genomics	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
3	Markov chains and sequence dependence	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
4	Hidden Markov models and dynamic programming	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
5	Mixtures, latent classes, and expectation maximization	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
6	Probabilistic alignment and evolutionary models	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
7	Inference, uncertainty, and numerical stability	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
8	Model criticism, population structure, and reproducibility	Explain the model, apply it to a concrete case, and defend the relevant correctness or design decision.
9	Integration workshop	Connect at least three chapters in one end-to-end problem and explain where their contracts meet.
10	Synthesis and project review	Audit assumptions and explain a complete solution from interface to failure handling.

Scope anchor: <https://catalog.registrar.ucla.edu/course/2025/COMSCIC121>. The sequence is aligned to the official catalog description and expanded into a practical ten-week study plan.

How I would use this packet

Treat each numbered section as one chapter. Read the formal explanation, pause at the student interpretation, and see whether the easy version still says the same thing. Rework the example without looking, then cover the solution in the chapter practice box. The cumulative set at the end is deliberately mixed: knowledge becomes durable when I have to choose the tool instead of being told which chapter I am in.

Contents

1	Genomic data as a probabilistic object	5
1.1	Rebuild the chapter from first principles	5
1.2	Details I would refuse to skip	5
1.3	How this chapter connects	6
1.4	What I need to be able to do	6
1.5	Deep notes	6
2	Conditional probability and Bayesian genomics	8
2.1	Rebuild the chapter from first principles	8
2.2	Details I would refuse to skip	8
2.3	How this chapter connects	9
2.4	What I need to be able to do	9
2.5	Deep notes	9
3	Markov chains and sequence dependence	11
3.1	Rebuild the chapter from first principles	11
3.2	Details I would refuse to skip	11
3.3	How this chapter connects	12
3.4	What I need to be able to do	12
3.5	Deep notes	12
4	Hidden Markov models and dynamic programming	13
4.1	Rebuild the chapter from first principles	14
4.2	Details I would refuse to skip	14
4.3	How this chapter connects	14
4.4	What I need to be able to do	15
4.5	Deep notes	15
5	Mixtures, latent classes, and expectation maximization	16
5.1	Rebuild the chapter from first principles	16
5.2	Details I would refuse to skip	17
5.3	How this chapter connects	17
5.4	What I need to be able to do	17
5.5	Deep notes	18

6	Probabilistic alignment and evolutionary models	19
6.1	Rebuild the chapter from first principles	19
6.2	Details I would refuse to skip	20
6.3	How this chapter connects	20
6.4	What I need to be able to do	20
6.5	Deep notes	20
7	Inference, uncertainty, and numerical stability	22
7.1	Rebuild the chapter from first principles	22
7.2	Details I would refuse to skip	22
7.3	How this chapter connects	23
7.4	What I need to be able to do	23
7.5	Deep notes	23
8	Model criticism, population structure, and reproducibility	25
8.1	Rebuild the chapter from first principles	25
8.2	Details I would refuse to skip	25
8.3	How this chapter connects	26
8.4	What I need to be able to do	26
8.5	Deep notes	26

CHAPTER 1 Genomic data as a probabilistic object

The idea

Genomic analysis begins with biological questions, sequence and variant representations, noisy measurement, population sampling, and a model for how observed data were generated. A probability model separates latent biological state from sequencing and sampling error.

Rebuild the chapter from first principles

The claim I need to own is this: Genomic analysis begins with biological questions, sequence and variant representations, noisy measurement, population sampling, and a model for how observed data were generated. A probability model separates latent biological state from sequencing and sampling error. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of genomic data as a probabilistic object. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach genomic data as a probabilistic object on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from the course foundations to conditional probability and bayesian genomics. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** Genomic analysis begins with biological questions, sequence and variant representations, noisy measurement, population sampling, and a model for how observed data were generated.
- **Mechanism.** A probability model separates latent biological state from sequencing and sampling error.
- **Boundary.** The useful test is whether I can apply genomic data as a probabilistic object to a new case and explain the assumption that makes the answer valid.
- **Decision test.** I should be able to say when genomic data as a probabilistic object is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

I write down what was observed, what is hidden, and which part of the story is noise before I touch an equation.

The easy version

Genomic analysis begins with biological questions, sequence and variant representations, noisy measurement, population sampling, and a model for how observed data were generated.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: Why is a reference genome not a probability

model? The conclusion is A reference supplies coordinates and a comparison sequence, but it does not specify variation, measurement error, ancestry, or sampling. A probabilistic model must assign likelihoods to possible observations under explicit latent states.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. Why is a reference genome not a probability model?

Solution. A reference supplies coordinates and a comparison sequence, but it does not specify variation, measurement error, ancestry, or sampling. A probabilistic model must assign likelihoods to possible observations under explicit latent states.

Practice 2. Give a short oral explanation of genomic data as a probabilistic object that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.

Solution. A strong answer begins with the model in the blue box, names the assumption that licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct genomic data as a probabilistic object from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.

Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a genomic data as a probabilistic object problem but never states a model or checks the result. Write the shortest useful review of that solution.

Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 2 Conditional probability and Bayesian genomics

The idea

Bayes' rule combines a prior over biological hypotheses with a likelihood for observed genomic data. Conditional independence reduces complicated joint distributions, while posterior odds make prior assumptions and evidence visible.

Rebuild the chapter from first principles

The claim I need to own is this: Bayes' rule combines a prior over biological hypotheses with a likelihood for observed genomic data. Conditional independence reduces complicated joint distributions, while posterior odds make prior assumptions and evidence visible. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of conditional probability and bayesian genomics. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach conditional probability and bayesian genomics on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from genomic data as a probabilistic object to markov chains and sequence dependence. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** Bayes' rule combines a prior over biological hypotheses with a likelihood for observed genomic data.
- **Mechanism.** Conditional independence reduces complicated joint distributions, while posterior odds make prior assumptions and evidence visible.
- **Boundary.** The useful test is whether I can apply conditional probability and bayesian genomics to a new case and explain the assumption that makes the answer valid.
- **Decision test.** I should be able to say when conditional probability and bayesian genomics is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

I use Bayes as bookkeeping for assumptions and evidence, not as a formula that makes weak data certain.

The easy version

Bayes' rule combines a prior over biological hypotheses with a likelihood for observed genomic data.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: A variant prior is 0.01 and a positive read

has sensitivity 0.99 and false-positive rate 0.05. Find the posterior after one positive read. The conclusion is The numerator is 0.99 times 0.01, or 0.0099. Total positive probability is 0.0099 plus 0.05 times 0.99, or 0.0594. The posterior is about 0.1667, showing why a rare prior still matters.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. A variant prior is 0.01 and a positive read has sensitivity 0.99 and false-positive rate 0.05. Find the posterior after one positive read.

Solution. The numerator is 0.99 times 0.01, or 0.0099. Total positive probability is 0.0099 plus 0.05 times 0.99, or 0.0594. The posterior is about 0.1667, showing why a rare prior still matters.

Practice 2. Give a short oral explanation of conditional probability and bayesian genomics that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.

Solution. A strong answer begins with the model in the blue box, names the assumption that licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct conditional probability and bayesian genomics from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.

Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a conditional probability and bayesian genomics problem but never states a model or checks the result. Write the shortest useful review of that solution.

Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 3 Markov chains and sequence dependence

The idea

A Markov model factors a sequence through local transition probabilities. Initial distribution, transition matrix, stationarity, order, and state meaning determine which dependence the model can express and which long-range structure it misses.

Rebuild the chapter from first principles

The claim I need to own is this: A Markov model factors a sequence through local transition probabilities. Initial distribution, transition matrix, stationarity, order, and state meaning determine which dependence the model can express and which long-range structure it misses. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of markov chains and sequence dependence. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach markov chains and sequence dependence on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from conditional probability and bayesian genomics to hidden markov models and dynamic programming. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** A Markov model factors a sequence through local transition probabilities.
- **Mechanism.** Initial distribution, transition matrix, stationarity, order, and state meaning determine which dependence the model can express and which long-range structure it misses.
- **Boundary.** The useful test is whether I can apply markov chains and sequence dependence to a new case and explain the assumption that makes the answer valid.
- **Decision test.** I should be able to say when markov chains and sequence dependence is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

I label each state as a biological claim, not just a matrix row; otherwise a clean computation can answer the wrong question.

The easy version

A Markov model factors a sequence through local transition probabilities.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: For a first-order DNA model, what conditional independence is assumed? The conclusion is Given the current base or state, the next base is independent of all earlier bases. The sequence probability factors into an initial probability

and one transition probability per adjacent pair.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. For a first-order DNA model, what conditional independence is assumed?

Solution. Given the current base or state, the next base is independent of all earlier bases. The sequence probability factors into an initial probability and one transition probability per adjacent pair.

Practice 2. Give a short oral explanation of markov chains and sequence dependence that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.

Solution. A strong answer begins with the model in the blue box, names the assumption that licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct markov chains and sequence dependence from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.

Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a markov chains and sequence dependence problem but never states a model or checks the result. Write the shortest useful review of that solution.

Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 4 Hidden Markov models and dynamic programming

The idea

A hidden Markov model adds latent states with transition probabilities and emissions for observations. Forward computes total likelihood, backward supplies suffix evidence, Viterbi finds the most probable path, and posterior decoding answers a different marginal question.

Rebuild the chapter from first principles

The claim I need to own is this: A hidden Markov model adds latent states with transition probabilities and emissions for observations. Forward computes total likelihood, backward supplies suffix evidence, Viterbi finds the most probable path, and posterior decoding answers a different marginal question. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of hidden markov models and dynamic programming. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach hidden markov models and dynamic programming on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from markov chains and sequence dependence to mixtures, latent classes, and expectation maximization. The earlier material supplies the language and constraints; this chapter

adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** A hidden Markov model adds latent states with transition probabilities and emissions for observations.
- **Mechanism.** Forward computes total likelihood, backward supplies suffix evidence, Viterbi finds the most probable path, and posterior decoding answers a different marginal question.
- **Boundary.** The useful test is whether I can apply hidden markov models and dynamic programming to a new case and explain the assumption that makes the answer valid.
- **Decision test.** I should be able to say when hidden markov models and dynamic programming is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

I keep 'best complete path' separate from 'best state at each position' because they are not interchangeable.

The easy version

A hidden Markov model adds latent states with transition probabilities and emissions for observations.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: Why can posterior decoding differ from the Viterbi path? The conclusion is Viterbi maximizes one joint state sequence. Posterior decoding maximizes each position's marginal state probability after summing over all paths, so independently best states need not form the globally best path.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. Why can posterior decoding differ from the Viterbi path?

Solution. Viterbi maximizes one joint state sequence. Posterior decoding maximizes each position's marginal state probability after summing over all paths, so independently best states need not form the globally best path.

Practice 2. Give a short oral explanation of hidden markov models and dynamic programming that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.

Solution. A strong answer begins with the model in the blue box, names the assumption that licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct hidden markov models and dynamic programming from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.

Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a hidden markov models and dynamic programming problem but never states a model or checks the result. Write the shortest useful review of that solution.

Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 5 Mixtures, latent classes, and expectation maximization

The idea

Mixture models represent heterogeneous samples through latent component assignments. Expectation maximization alternates posterior responsibilities with parameter updates, increases likelihood locally, and requires initialization and identifiability checks.

Rebuild the chapter from first principles

The claim I need to own is this: Mixture models represent heterogeneous samples through latent component assignments. Expectation maximization alternates posterior responsibilities with parameter updates, increases likelihood locally, and requires initialization and identifiability checks. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of mixtures, latent classes, and expectation maximization. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach mixtures, latent classes, and expectation maximization on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from hidden markov models and dynamic programming to probabilistic alignment and evolutionary models. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure

would make me switch approaches.

Deep notes

- **Model.** Mixture models represent heterogeneous samples through latent component assignments.
- **Mechanism.** Expectation maximization alternates posterior responsibilities with parameter updates, increases likelihood locally, and requires initialization and identifiability checks.
- **Boundary.** The useful test is whether I can apply mixtures, latent classes, and expectation maximization to a new case and explain the assumption that makes the answer valid.
- **Decision test.** I should be able to say when mixtures, latent classes, and expectation maximization is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

I treat a discovered cluster as a model component until biology independently supports the label.

The easy version

Mixture models represent heterogeneous samples through latent component assignments.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: What does the E-step compute in a two-component mixture? The conclusion is For each observation it computes posterior membership probabilities using current mixing weights and component likelihoods. Those fractional responsibilities become weights in the M-step.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. What does the E-step compute in a two-component mixture?

Solution. For each observation it computes posterior membership probabilities using current mixing weights and component likelihoods. Those fractional responsibilities become weights in the M-step.

Practice 2. Give a short oral explanation of mixtures, latent classes, and expectation maximization that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.

Solution. A strong answer begins with the model in the blue box, names the assumption that licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a

four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct mixtures, latent classes, and expectation maximization from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.

Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a mixtures, latent classes, and expectation maximization problem but never states a model or checks the result. Write the shortest useful review of that solution.

Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 6 Probabilistic alignment and evolutionary models

The idea

Sequence comparison needs a model for substitutions, insertions, deletions, background composition, and evolutionary distance. Log-odds scoring compares a related-sequence model with a null model, while dynamic programming organizes alignments.

Rebuild the chapter from first principles

The claim I need to own is this: Sequence comparison needs a model for substitutions, insertions, deletions, background composition, and evolutionary distance. Log-odds scoring compares a related-sequence model with a null model, while dynamic programming organizes alignments. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of probabilistic alignment and evolutionary models. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.

4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach probabilistic alignment and evolutionary models on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from mixtures, latent classes, and expectation maximization to inference, uncertainty, and numerical stability. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** Sequence comparison needs a model for substitutions, insertions, deletions, background composition, and evolutionary distance.
- **Mechanism.** Log-odds scoring compares a related-sequence model with a null model, while dynamic programming organizes alignments.
- **Boundary.** The useful test is whether I can apply probabilistic alignment and evolutionary models to a new case and explain the assumption that makes the answer valid.

- **Decision test.** I should be able to say when probabilistic alignment and evolutionary models is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

My shorthand for this chapter is: an alignment score only means something relative to the model that produced its match, mismatch, and gap terms.

The easy version

Sequence comparison needs a model for substitutions, insertions, deletions, background composition, and evolutionary distance.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: Why use log-odds substitution scores? The conclusion is A log likelihood ratio adds across positions and rewards pairs more likely under the related model than under background. Positive values support relatedness and negative values support the null.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. Why use log-odds substitution scores?

Solution. A log likelihood ratio adds across positions and rewards pairs more likely under the related model than under background. Positive values support relatedness and negative values support the null.

Practice 2. Give a short oral explanation of probabilistic alignment and evolutionary models that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.

Solution. A strong answer begins with the model in the blue box, names the assumption that licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct probabilistic alignment and evolutionary models from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.

Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a probabilistic alignment and evolutionary models problem but never states a model or checks the result. Write the shortest useful review of that solution.

Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 7 Inference, uncertainty, and numerical stability

The idea

Maximum likelihood, posterior inference, intervals, likelihood ratios, permutation tests, and multiple-testing control answer different uncertainty questions. Log probabilities, log-sum-exp, and scaled recursions prevent underflow.

Rebuild the chapter from first principles

The claim I need to own is this: Maximum likelihood, posterior inference, intervals, likelihood ratios, permutation tests, and multiple-testing control answer different uncertainty questions. Log probabilities, log-sum-exp, and scaled recursions prevent underflow. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of inference, uncertainty, and numerical stability. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.

- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach inference, uncertainty, and numerical stability on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from probabilistic alignment and evolutionary models to model criticism, population structure, and reproducibility. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** Maximum likelihood, posterior inference, intervals, likelihood ratios, permutation tests, and multiple-testing control answer different uncertainty questions.
- **Mechanism.** Log probabilities, log-sum-exp, and scaled recursions prevent underflow.
- **Boundary.** The useful test is whether I can apply inference, uncertainty, and numerical stability to a new case and explain the assumption that makes the answer valid.
- **Decision test.** I should be able to say when inference, uncertainty, and numerical stability is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

I report uncertainty beside the biological conclusion because a single best call hides how fragile the evidence may be.

The easy version

Maximum likelihood, posterior inference, intervals, likelihood ratios, permutation tests, and multiple-testing control answer different uncertainty questions.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: Why do forward probabilities underflow on long sequences? The conclusion is They multiply many numbers below one and eventually round to zero. Work in log space with log-sum-exp or rescale each recursion step while tracking the scale factors.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. Why do forward probabilities underflow on long sequences?

Solution. They multiply many numbers below one and eventually round to zero. Work in log space with log-sum-exp or rescale each recursion step while tracking the scale factors.

Practice 2. Give a short oral explanation of inference, uncertainty, and numerical stability that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.

Solution. A strong answer begins with the model in the blue box, names the assumption that licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct inference, uncertainty, and numerical stability from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.

Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a inference, uncertainty, and numerical stability problem but never states a model or checks the result. Write the shortest useful review of that solution.

Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

CHAPTER 8 Model criticism, population structure, and reproducibility

The idea

Genomic models can fail through population structure, batch effects, relatedness, reference bias, leakage, selection bias, and incorrect independence assumptions. Reproducible analysis records provenance, software, parameters, uncertainty, and subgroup behavior.

Rebuild the chapter from first principles

The claim I need to own is this: Genomic models can fail through population structure, batch effects, relatedness, reference bias, leakage, selection bias, and incorrect independence assumptions. Reproducible analysis records provenance, software, parameters, uncertainty, and subgroup behavior. I do not count that as learned just because the sentence looks familiar. I should be able to name the objects, state what changes and what stays fixed, recover the rule from the definitions, and explain why the result is allowed.

The move

My working sequence

1. **Translate.** Rewrite the prompt in the language of model criticism, population structure, and reproducibility. List the known quantities, the unknown, the units or types, and any hidden constraint.
2. **Choose.** State the theorem, model, algorithm, accounting rule, or physical law before substituting values. This is where I make the assumption visible.
3. **Execute.** Work one justified step at a time and keep intermediate state readable enough that I can audit it later.
4. **Verify.** Check a boundary case, sign, unit, invariant, order of magnitude, or alternative derivation. Then translate the result back into an ordinary sentence.

Details I would refuse to skip

- **Definitions:** I should be able to define every technical noun in the chapter without using that same noun in the definition.
- **Assumptions:** I write the condition that makes the method legal beside the method itself. A correct procedure under the wrong assumptions is still a wrong answer.
- **Representation:** I move freely among words, equations or code, a diagram, and a small numerical example. If one representation disagrees with another, that disagreement is useful evidence.
- **Failure modes:** I keep the nearest counterexample in mind: the boundary where the rule changes, the invariant breaks, or the model stops matching reality.
- **Communication:** My final answer names what the result means and what it does not establish. I do not let notation do the explanatory work for me.

Sanity check

Closed-note check. Can I teach model criticism, population structure, and reproducibility on a blank page, derive or reconstruct its main rule, work a fresh example, and diagnose a deliberately broken solution? If I can only recognize the finished work, I am not done.

How this chapter connects

I read this chapter as the bridge from inference, uncertainty, and numerical stability to the cumulative course model. The earlier material supplies the language and constraints; this chapter adds a new reasoning move; the later material asks me to use that move without being told its name. That connection matters because exams and real projects mix chapters on purpose.

What I need to be able to do

For this chapter, I should be able to define the objects and state involved, state the governing invariant or contract, trace a small example without hand-waving, and explain which assumption makes the method valid. I should also be able to compare this model with the nearest alternative and say what failure would make me switch approaches.

Deep notes

- **Model.** Genomic models can fail through population structure, batch effects, relatedness, reference bias, leakage, selection bias, and incorrect independence assumptions.
- **Mechanism.** Reproducible analysis records provenance, software, parameters, uncertainty, and subgroup behavior.
- **Boundary.** The useful test is whether I can apply model criticism, population structure, and reproducibility to a new case and explain the assumption that makes the answer valid.
- **Decision test.** I should be able to say when model criticism, population structure, and reproducibility is the right model, when its assumptions fail, and what evidence would change my choice.

How I actually think about it

My shorthand for this chapter is: a tiny p-value does not rescue a biased sample or a model that confused ancestry with causality.

The easy version

Genomic models can fail through population structure, batch effects, relatedness, reference bias, leakage, selection bias, and incorrect independence assumptions.

Worked example

A useful way to work this chapter is to make the smallest concrete instance, write the state or contract explicitly, and then scale the reasoning only after that instance behaves correctly. The worked question below makes that reasoning concrete: A classifier predicts disease well only within

one ancestry group. What should be checked? The conclusion is Audit sampling, prevalence, population structure, calibration and error by subgroup, batch and site effects, leakage, and transport to external cohorts. Restrict claims until independent data support broader use.

Common miss

The common mistake is to memorize the visible procedure while dropping its precondition. I write the assumption beside the method and test one boundary case before trusting the result.

Solved chapter practice

Practice. A classifier predicts disease well only within one ancestry group. What should be checked?

Solution. Audit sampling, prevalence, population structure, calibration and error by subgroup, batch and site effects, leakage, and transport to external cohorts. Restrict claims until independent data support broader use.

Practice 2. Give a short oral explanation of model criticism, population structure, and reproducibility that includes its model, one assumption, one failure mode, and one comparison with a neighboring method.

Solution. A strong answer begins with the model in the blue box, names the assumption that licenses the method, uses the common miss above as the failure mode, and chooses the comparison according to the decision test in the deep notes. The goal is not one memorized sentence; it is a four-part explanation I can reproduce on a new example.

Mastery practice A. Reconstruct model criticism, population structure, and reproducibility from a blank page. Include the central definition, the governing rule, one required assumption, and one boundary case.

Solution. Start from the blue idea box, but rewrite it in my own words. Identify the objects and state, name the rule before using it, copy the relevant assumption from the details audit, and choose a boundary where that assumption becomes equality or fails. A complete solution explains why each part belongs; a list of vocabulary is not enough.

Mastery practice B. A classmate gets a polished-looking answer to a model criticism, population structure, and reproducibility problem but never states a model or checks the result. Write the shortest useful review of that solution.

Solution. I would ask four concrete questions: What exactly are the inputs and desired output? Which rule licenses the main step? Which assumption could invalidate it? Which independent check supports the conclusion? If the answer cannot survive those four questions, I treat the numerical or verbal result as unverified.

Final study check

I should be able to close this packet and reconstruct the core models, not merely recognize their names. My final check is to explain one complete problem aloud: what the inputs mean, which invariant or contract controls the work, why the method is legal, what it costs, how it can fail, and what evidence would convince another engineer.