AI for Genomic Science

Contents

AI for Genomic Science 3
- About This Book . . . 4
- What Makes This Book Different? . . . 4
  - □ Biology–First Approach . . . 4
  - □ Hands–On Coding Labs . . . 4
  - □ Clear Mathematical Explanations . . . 4
  - □ Real Case Studies . . . 4
- What You'll Learn . . . 4
- Book Structure . . . 5
  - Part 1: Foundations of Artificial Intelligence . . . 5
  - Part 2: Genetic Variants and Early AI Approaches . . . 5
  - Part 3: Deep Learning for Genomic Sequence Analysis . . . 5
  - Part 4: Language Models Meet DNA . . . 5
  - Part 5: Single–Cell Omics and Foundation Models . . . 5
- How to Use This Book . . . 5
  - For Self–Study: . . . 5
  - For Classroom Use: . . . 5
  - Prerequisites: . . . 6
- Getting Started . . . 6
  - Setting Up Your Environment . . . 6
- Quick Navigation . . . 6
- License . . . 7

Chapter 1: Why AI for Genomic Science? 7
- The Biological Challenge . . . 7
- The Hierarchy of Artificial Intelligence (AI): From Broad to Specific . . . 8
  - Machine Learning (ML) . . . 8
  - Deep Learning (DL) . . . 8
  - The Hierarchy in Practice . . . 9
- Correlation vs. Causation: What AI Can and Cannot Tell You . . . 9
  - The Fundamental Limitation . . . 9
  - A Concrete Example: Gene Expression and Disease . . . 9
  - Establishing Causation Requires Molecular Experiments . . . 10
- Causal Inference: A Brief Introduction . . . 11
- From Null Hypothesis Testing to Comparing Causal Models . . . 11
- Success Stories: AI Transforming Genomics . . . 12
  - Protein Structure: AlphaFold (2020–2024) . . . 12
  - Variant Calling and Interpretation (2015–present) . . . 12
  - Single–Cell Analysis: Foundation Models (2022–2024) . . . 12
  - Drug Discovery: Virtual Screening (2020–present) . . . 12
  - Common Success Patterns . . . 12
- The Paradigm Shift: How AI Changes Biology . . . 12

From Linear to Iterative Discovery . . . 13
The Virtual Cell Vision . . . 13
Active Learning: Scientists in the Loop . . . 14
What Biologists Need to Know About AI . . . 14
Practical Skill Levels . . . 14
When to Use AI . . . 14
Common AI Failures and Lessons . . . 15
Summary . . . 15
Key Terms . . . 16
Test Your Understanding: Can You Answer These? . . . 16
Hands–On Labs . . . 17
Lab 1.1: Getting Started with Google Colab and Python (30–45 min) . . . 17
Lab 1.2: Essential Python Tools for Bioinformatics (60–90 min) . . . 17

Chapter 2: From Biological Intuition to Deep Learning **17**
Your Brain Already Does Bayesian Inference . . . 18
You're a Natural Statistician . . . 18
The Bayesian Formula (Don't Panic!) . . . 21
Biology Is Fundamentally Probabilistic . . . 21
What Does This Mean? . . . 22
Example 1: Gene Expression is a Distribution, Not a Number . . . 22
Example 2: Protein Folding is an Ensemble, Not a Structure . . . 23
Example 3: Evolution is Bayesian Updating Over Generations . . . 23
Example 4: Your Immune System Learns Like a Bayesian . . . 26
The Pattern: Biology = Probability Distributions . . . 26
The Globe–Tossing Experiment: Learning from Data . . . 27
The Setup . . . 27
The Naïve Answer . . . 27
Considering Different Hypotheses . . . 27
Before Any Data: The Prior . . . 27
After Seeing Data: The Likelihood . . . 28
Calculating the Posterior . . . 29
Visualizing Bayesian Updating . . . 29
The Power of More Data . . . 31
The Bayesian Formulation . . . 31
From Bayesian Inference to Deep Learning . . . 32
The Ideal vs. The Practical . . . 32
Understanding MAP: Back to the Globe . . . 33
The Mathematical Connection: Loss Functions . . . 34
Training = Climbing to the Peak . . . 36
One More Thing: Preferring Simpler Models . . . 37
Summary: The Bayesian–Deep Learning Dictionary . . . 38
Why Not Keep the Full Distribution? . . . 38
Why This Connection Matters for Genomics . . . 39
Interpret Model Confidence Correctly . . . 39
Understand Why More Data Helps . . . 39
Recognize When Models Fail . . . 40
Design Better Experiments . . . 40
Math Box: Bayes' Theorem and Components . . . 41
The Complete Formula . . . 41
Calculating Evidence (Normalization) . . . 41
Binomial Likelihood . . . 41
Updating with Each Observation . . . 42
Summary . . . 42

Key Terms 43
Test Your Understanding: Can You Answer These? 43
Coding Lab 2: Bayesian Inference with Globe-Tossing 45

AI for Genomic Science

Welcome to **AI for Genomic Science**

Author: Joon-Yong An, Korea University

Last Update: 2025/11/2 (Under Construction - probably weekly update by 2026 if I am not lazy enough..)

Figure 1: Front Cover

About This Book

This textbook introduces how artificial intelligence is revolutionizing biological research — from analyzing genetic variants to modeling entire cells. It is designed specifically for undergraduate biology students (senior level) who want to understand the computational approaches that are transforming genomics, without requiring prior experience in AI or advanced programming.

The field of genomics has been rapidly transformed by machine learning, deep learning, and large-scale computational methods. These advances now allow us to analyze massive genomic datasets, predict functional impacts of variants, and model complex biological systems with unprecedented accuracy. This textbook takes a biology-first approach to cover the essential AI concepts and methods that undergraduate students need to master, integrating computational techniques with genomic applications.

This book is written for **biology majors** who have a solid foundation in molecular biology and genetics, are curious about computational approaches, want to understand the **why** and **how** behind AI methods in genomics, and are comfortable with basic mathematics. Rather than assuming extensive programming background, we start from the basics and build up gradually, emphasizing conceptual understanding alongside practical applications.

Please note that chapters are currently being written and improved. The complete version is expected to be finished by 2026!

What Makes This Book Different?

☐ **Biology-First Approach**

Each chapter starts with a **real biological challenge**—experimental limitations that motivate computational solutions. You'll never wonder "why do I need to learn this?"

☐ **Hands-On Coding Labs**

Every chapter includes **Google Colab-based coding exercises**. No installation needed—just click and start learning! All code is heavily commented and designed for beginners.

☐ **Clear Mathematical Explanations**

Math concepts are explained in **Math Boxes** with biological examples. We won't shy away from equations, but we'll make sure you understand what they mean.

☐ **Real Case Studies**

Learn from actual research papers and real datasets. See how these methods are being used to make biological discoveries right now.

What You'll Learn

By the end of this book, you will be able to:

☐ Understand the fundamental concepts of machine learning and deep learning
☐ Explain how AI methods predict the effects of genetic variants
☐ Use pre-trained models to analyze genomic sequences

☐ Interpret results from tools like CADD, DeepSEA, Enformer, and DNABERT
☐ Understand how language models are applied to DNA and RNA sequences
☐ Analyze single-cell omics data using foundation models
☐ Critically evaluate AI-based studies in genomics literature
☐ Write basic Python code for bioinformatics analyses

Book Structure

This textbook is organized into **five parts**:

Part 1: Foundations of Artificial Intelligence

Learn the essential AI concepts every biologist should know—from neural networks to different architectures.

Part 2: Genetic Variants and Early AI Approaches

Understand how traditional and machine learning methods help us prioritize and interpret genetic variants.

Part 3: Deep Learning for Genomic Sequence Analysis

Explore how convolutional neural networks and transformers predict regulatory elements and variant effects from DNA sequences.

Part 4: Language Models Meet DNA

Discover how natural language processing techniques are revolutionizing genomics through DNA language models and foundation models.

Part 5: Single-Cell Omics and Foundation Models

See how AI is helping us understand individual cells and move toward whole-cell computational models.

How to Use This Book

For Self-Study:

1. Read each chapter sequentially—concepts build on each other
2. Work through the **Coding Labs** in Google Colab
3. Try the **Discussion Questions** to deepen your understanding
4. Explore the **Further Reading** for topics that interest you

For Classroom Use:

- Each chapter is designed for 1–2 weeks of instruction
- Coding labs can be used as homework or in-class activities
- Discussion questions are great for group work
- Case studies make excellent presentation topics

Prerequisites:

- **Biology**: Molecular biology, genetics (sophomore/junior level)
- **Math**: Algebra, basic statistics (we'll review when needed)
- **Programming**: None required! We start from scratch
- **Enthusiasm**: Essential! □

Getting Started

Setting Up Your Environment

All coding exercises use **Google Colab**, which runs in your web browser. You'll need: 1. A Google account (free) 2. Internet connection 3. That's it!

No software installation required. We'll walk you through everything in Chapter 1.

Quick Navigation

Part 1: Foundations of AI

- Chapter 1: Why AI for Genomic Science?
- Chapter 2: From Biological Intuition to Deep Learning
- [Chapter 3: Neural Networks Basics (Soon!)]
- [Chapter 4: Types of Neural Networks (Soon!)]

Part 2: Genetic Variants and Early AI

- [Chapter 5: Genetic Variation and Genomic Technologies (Soon!)]
- [Chapter 6: Evolutionary Conservation and Traditional Tools (Soon!)]
- [Chapter 7: Machine Learning Ensemble Methods (Soon!)]

Part 3: Deep Learning for Genomic Sequences

- [Chapter 8: The Rise of Deep Learning in Genomics (Soon!)]
- [Chapter 9: CNN–Based Regulatory Sequence Analysis (Soon!)]
- [Chapter 10: Transformer–Based Models (Soon!)]

Part 4: Language Models Meet DNA

- [Chapter 11: Introduction to Language Models (Soon!)]
- [Chapter 12: Foundation Models for Genomics (Soon!)]
- [Chapter 13: DNA Language Models (Soon!)]
- [Chapter 14: Next–Generation DNA Models (Soon!)]

Part 5: Single–Cell Omics and Foundation Models

- [Chapter 15: Introduction to Single–Cell Omics (Soon!)]
- [Chapter 16: Single–Cell Foundation Models (Soon!)]
- [Chapter 17: Toward Whole–Cell Modeling (Soon!)]

Happy Learning! □□

License

License information to be added

Chapter 1: Why AI for Genomic Science?

Dr. Sarah An stares at her computer screen, frustrated. She just received whole-genome sequencing (WGS) data from a 7-year-old patient with a complex rare disorder—severe skeletal abnormalities, neurodevelopmental delays, and metabolic dysfunction. Both parents are healthy, suggesting this is a de novo (new) mutation. The WGS reveals 4.5 million genetic variants compared to the reference genome. Her computational pipeline identifies approximately 70 de novo variants—changes found in the patient but not in either parent. These are the prime suspects. She runs standard pathogenicity prediction tools—CADD scores and PolyPhen-2—to narrow down the list. After filtering, she's left with 3 coding mutations** in genes with unknown function or disease relevance, and 7 noncoding variants** in regulatory regions.

Any of these 10 variants could be causative. But which one?

Each functional validation experiment takes 2–3 months and costs $8,000–15,000. Testing all 10 would take **2+ years** and over **$100,000**. Even then, the noncoding variants are challenging to test—their effects on gene regulation are subtle and context-dependent, requiring cell-type-specific assays, enhancer reporter experiments, and potentially CRISPR editing in patient-derived cells. Three days later, using an AI-powered variant prioritization model trained on millions of variants and functional genomics data, Sarah narrows the list to **2 high-confidence candidates**: one coding variant in a gene involved in skeletal development with a damaging structural prediction, and one noncoding variant in an enhancer predicted to disrupt binding of a critical transcription factor expressed in developing bone and neural tissue. Within two months, functional experiments confirm the noncoding variant as causative—it disrupts an enhancer driving expression of a gene essential for both skeletal and neural development.

This is one example how we can leverage the power of AI in genomics: not replacing scientific intuition, but dramatically amplifying it to navigate the vast search space of human genetic variation.

The Biological Challenge

Modern biology faces two unprecedented explosions.

Data Explosion: Whole genome sequencing can easily read a single human genome of approximately 3 billion nucleotides, with each person having 3–5 million variants that differ from one another. Single-cell RNA-seq experiments generate data from millions of cells, while various epigenome sequencing techniques can provide information for chromatin accessibility maps across millions of genomic regions. Proteomics technologies can identify hundreds of millions of peptide sequences from a single experiment.

Many Hypothesis to be explored: Large-scale genomics studies don't just generate data but they also generate thousands of testable hypotheses. GWAS studies identify hundreds of loci associated with each complex trait, but most are in noncoding regions with unknown mechanisms. Genome sequencing in rare disease cohorts reveals dozens of candidate genes per patient, each requiring functional validation. Cancer genomics finds hundreds of somatic mutations per tumor, but only a subset are "driver" mutations versus neutral "passengers." Single-cell atlases reveal thousands of cell type-specific gene expression patterns, each suggesting regulatory hypotheses. Spatial transcriptomics shows genes co-expressed in tissue neighborhoods, implicating thousands of potential cell-cell interactions.

Systems-Level Complexity: The challenge isn't just quantity but the complexity inherent in our cells and tissues. Genes operate in networks, not isolation, and a single phenotype often involves dozens to hundreds of genes working together. Context matters profoundly: the same variant can be benign in one genetic background but pathogenic in another, and gene function depends on cell type, developmental stage, and environmental conditions. Combinatorial interactions add another layer of complexity, where

two variants individually benign might be harmful together, causing the number of possible combinations to explode exponentially. Pleiotropic effects further complicate the picture: one gene affects multiple phenotypes while one phenotype is affected by multiple genes, creating a many–to–many mapping rather than a simple one–to–one relationship.

The fundamental problem is that we can generate biological data and hypotheses far faster than we can experimentally test them. A single GWAS might implicate 500 genes, and testing each would take decades. AI helps us predict which experiments to prioritize and which hypotheses are most likely to be true.

The Hierarchy of Artificial Intelligence (AI): From Broad to Specific

Artificial Intelligence is the broadest concept—any technique that enables computers to mimic human intelligence. This includes playing chess, recognizing faces, translating languages, predicting protein structures, and classifying cell types.

Machine Learning (ML)

Machine Learning is a subset of AI focused on one question:

> "Can computers learn patterns from data instead of having humans program every rule explicitly?"

Instead of telling a computer "if the DNA sequence has TATA box at position –25 and GC content 〉60%, it's probably a promoter," we give the computer thousands of examples of promoters and non–promoters, and let it figure out the patterns. The learning process works by having the algorithm compare its predictions to the correct answers, measure the error, and then automatically adjust its approach through mathematical optimization.

What Is "Learning" in Machine Learning? Here is an example for "Predicting variant functional impact".

You have: – **Input:** Conservation score, population frequency, predicted structural change – **Output:** Likely functional impact or neutral

The algorithm learns by:

1. **Making initial guesses** (usually random)
2. **Comparing to known answers** (labeled training data)
3. **Measuring error** (calculating "loss")
4. **Adjusting internal parameters** to reduce error
5. **Repeating thousands of times**

This adjustment process is optimization—finding the best parameters that minimize prediction errors.

For a simple model, the algorithm discovers the best weights:

```
Score = (Conservation × w ) + (Frequency × w ) + (StructuralChange × w ) + bias
```

The algorithm discovers the best values for w_1, w_2, w_3, and bias by examining thousands of examples. We don't tell the algorithm what values to use—it discovers them from data.

Deep Learning (DL)

Deep Learning uses **artificial neural networks** with many layers to automatically discover patterns in data. Each layer builds on the previous one in a hierarchical fashion. The first layer might detect simple sequence motifs such as TATA or CAAT boxes, while the second layer might combine these motifs to detect larger regulatory modules. The third layer might identify context–dependent regulatory logic, and the fourth layer might predict cell–type–specific enhancer activity.

The Hierarchy in Practice

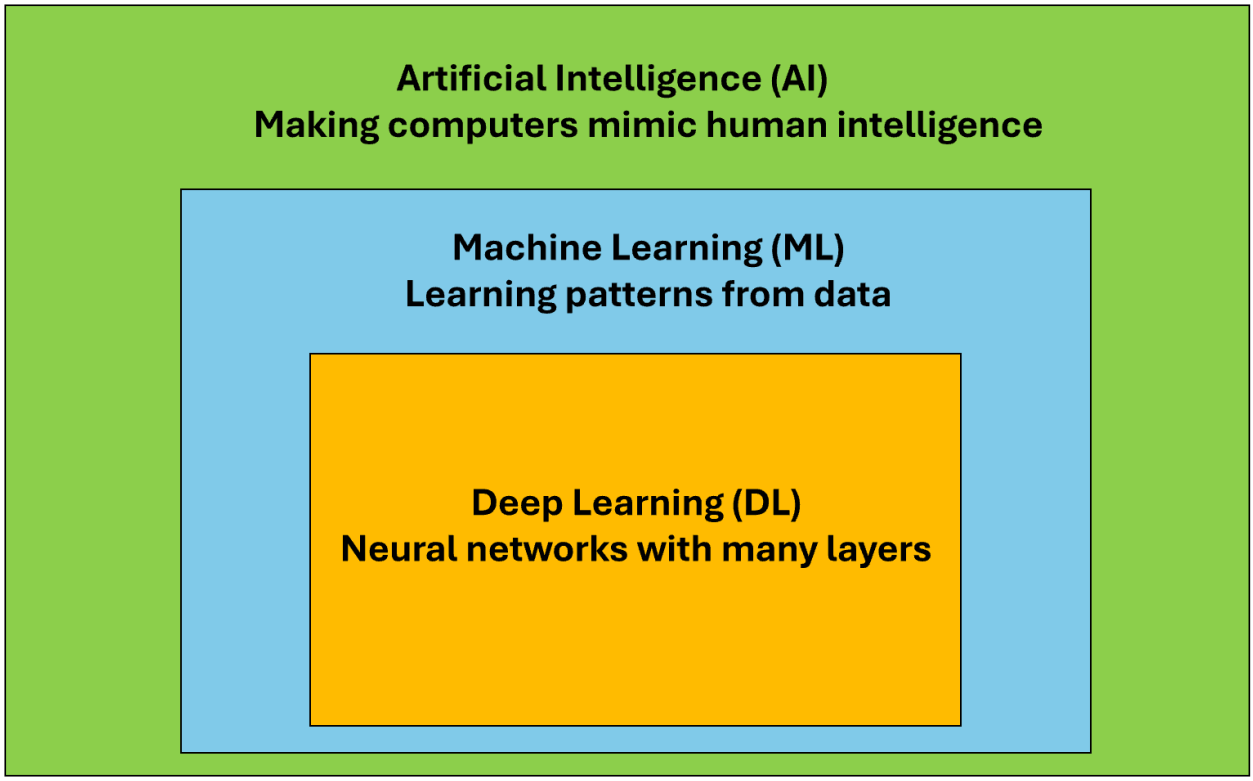

Figure 2: The AI Hierarchy

Figure 1.1: The AI Hierarchy – From Broad to Specific.

Genomic Tool Examples:

Tool	Category	Why?
BLAST	AI (not ML)	Uses programmed rules for alignment
Random Forest classifier	ML (not DL)	Learns from data, no neural networks
AlphaFold	DL	Deep neural networks with many layers

Correlation vs. Causation: What AI Can and Cannot Tell You

The Fundamental Limitation

AI models learn associations (correlations). They do not learn causation.

This is perhaps the most critical concept for biologists to understand.

A Concrete Example: Gene Expression and Disease

Suppose you have data showing: – Gene X expression is highly correlated with Disease Y – Correlation coefficient $r = 0.85$, $p < 0.001$

What can you conclude?

× **WRONG:** "Gene X causes Disease Y"

× **WRONG:** "Targeting Gene X will cure Disease Y"

□ **CORRECT:** "Gene X expression and Disease Y are associated"

Why? Consider these scenarios:

```
Scenario 1: Gene X → Disease Y (causal)
            [Targeting Gene X might cure disease]

Scenario 2: Disease Y → Gene X (reverse causation)
            [Gene X is just responding to disease]

Scenario 3: Inflammation → Gene X

                        Disease Y (confounding)
            [Both are symptoms; treat inflammation instead]

Scenario 4: Gene X ← Environmental Factor → Disease Y (common cause)
            [Change environment, not the gene]
```

All four scenarios produce identical correlations, but require completely different interventions!

> "The causes of the data cannot be extracted from the data alone. We need an additional external model, a causal model of some kind." — Richard McElreath, Statistical Rethinking

This is why Nancy Cartwright's slogan is so important: "No causes in, no causes out." You cannot discover causation by data mining alone—you must bring causal assumptions to the data.

Establishing Causation Requires Molecular Experiments

Things to prove causation:

1. Controlled perturbation
   - CRISPR knockout/knockdown: Use CRISPR gene editing to delete or reduce expression of the target gene, then check if the predicted phenotype appears
   - Drug inhibition: Apply a chemical compound that blocks the protein's function, then verify if the phenotype changes as expected
   - Overexpression: Artificially increase the gene's expression level using expression vectors, then observe if this enhances or triggers the predicted phenotype
2. Observation of effect
   - Phenotype changes?: Measure whether the observable characteristic (cell growth, morphology, disease marker) actually changes after perturbation
   - In predicted direction?: Confirm the change matches your hypothesis—if you predicted growth reduction, does growth actually decrease?
   - In the right context?: Verify the effect occurs in the relevant cell type, developmental stage, or environmental condition predicted by your model
3. Mechanism validation
   - How does it work?: Use biochemical assays, imaging, or sequencing to identify the molecular steps—does the gene regulate transcription, bind specific proteins, or alter signaling pathways?
   - Is the pathway what you expected?: Compare the observed molecular mechanism to your predicted pathway—do the right proteins interact, does the right signaling cascade activate?

AI's role: Predict which of 1000 genes to perturb first

Experiments' role: Establish that perturbation actually causes the effect

Causal Inference: A Brief Introduction

Modern causal inference uses **Directed Acyclic Graphs (DAGs)** to represent causal relationships. A DAG is a diagram with arrows showing cause–and–effect relationships, where "directed" means arrows have direction (A→B means A causes B) and "acyclic" means no circular loops exist (no A→B→C→A).

```
Simple DAG Example: Gene Regulation
    Transcription Factor (TF)
           ↓
       Enhancer Activity
           ↓
      Gene Expression
           ↓
        Protein Level
           ↓
        Phenotype

This DAG states:
- Changing TF changes Enhancer (causal)
- Changing Protein changes Phenotype (causal)
- Protein and TF are correlated, but not directly causally related
```

Key insight: The DAG helps you design experiments to establish causation, not just correlation.

For biology students: – Learn to draw causal models of your system – Use them to design experiments – Don't confuse AI predictions (correlations) with causal claims

---

From Null Hypothesis Testing to Comparing Causal Models

Many biology students learn statistics through a flowchart approach:

```
Is data normal? → YES → t-test
                → NO → Mann-Whitney U test
```

The goal is typically to reject a null hypothesis: "Does this gene variant have NO effect on disease risk?" If $p < 0.05$, we conclude "yes, it has an effect."

This approach has serious limitations in modern genomics:

1. **"No effect" is not a realistic biological hypothesis**
   - Every variant affects something, even if weakly
   - The real question isn't "effect or no effect?" but "what mechanism causes the effect?"
2. **Null models are not unique**
   - What's the "null" for gene network evolution?
   - What's the "null" for cell–cell communication?
   - Multiple process models can produce identical null predictions
3. **Industrial vs. research contexts**
   - The t–test was invented for Guinness Brewery quality control: "Is this batch the same as previous batches?"
   - But genomics research asks: "Which of these 10 mechanisms explains this phenotype?"

What this means for AI and genomics:

When an AI model predicts "this variant is 95% likely to be pathogenic," it's not just saying "effect exists." It's implicitly proposing mechanisms based on patterns it learned. Your job as a scientist is to: –

Formulate competing causal models (Does this variant disrupt transcription factor binding? Alter splicing? Change enhancer activity?) – Design experiments to compare these models – Use data to determine which mechanism best explains the phenotype

AI models should help you compare competing biological hypotheses, not just confirm that "something is significant."

Success Stories: AI Transforming Genomics

AI has already made real discoveries across genomics. Here are key examples:

Protein Structure: AlphaFold (2020–2024)

AlphaFold 2 achieved near–experimental accuracy predicting 3D protein structures in hours instead of months (Jumper et al 2021, Nature). AlphaFold 3 extended to protein complexes and DNA/RNA interactions (Abramson et al 2024, Nature). Over 200 million structures are now freely available, accelerating drug discovery and disease research.

Variant Calling and Interpretation (2015–present)

DeepVariant treats sequencing as image recognition, reducing error rates by 50% vs. traditional methods (Poplin et al 2018, Nature Biotechnology). Now standard in clinical sequencing. Models like DeepSEA and Basenji extended this to predict regulatory variant effects (Zhou & Troyanskaya 2015, Nature Methods; Kelley et al 2018, Genome Research). Transformer models predict gene expression, chromatin state, and histone modifications from DNA sequence alone (Avsec et al 2021, Nature Methods). This enables predicting noncoding variant effects and revealing long–range regulatory interactions up to 100kb away.

Single–Cell Analysis: Foundation Models (2022–2024)

Models like scGPT and Geneformer treat genes as words in language, learning universal cellular representations (Cui et al 2024, Nature Methods; Theodoris et al 2023, Nature). This enabled the Human Cell Atlas and reduced cell type annotation from weeks to hours.

Drug Discovery: Virtual Screening (2020–present)

Deep learning screens 100+ million molecules virtually in days. Halicin—a novel antibiotic effective against drug–resistant bacteria—was discovered this way (Stokes et al 2020, Cell). A major regulatory milestone came in 2024 when the FDA accepted Recursion Pharmaceuticals' AI–based models as a replacement for animal testing in certain toxicology studies. This represents the first time AI predictions were formally approved to substitute traditional animal experiments in drug development, potentially accelerating timelines while reducing costs and ethical concerns.

Common Success Patterns

These breakthroughs share key features: – **Massive datasets** for training – **Clear, measurable goals** – **Too expensive/slow** for comprehensive experimental testing – **Complex patterns** difficult to program explicitly – **Predictions guide** rather than replace experiments – **Rigorous experimental validation** of key predictions

---

The Paradigm Shift: How AI Changes Biology

Perhaps the most profound change AI brings isn't speed or scale—it's a fundamental transformation in how we do science.

From Linear to Iterative Discovery

Traditional: Hypothesis-Driven Research

```
Observe → Hypothesis → Experiment → Data → Accept/Reject → New Hypothesis
```

Limitations: One hypothesis at a time, months to years per cycle, testing only what we suspect.

New: AI-Augmented Discovery Loop

```
Large-scale data → AI training → Thousands of predictions
        ↑                                    ↓
    New data ← Selective validation ← Prioritize by confidence
```

From sequential testing to parallel exploration. AI generates thousands of hypotheses simultaneously, experiments validate the most promising ones, results improve the model, and the cycle accelerates.

The Virtual Cell Vision

Recent work envisions **AI Virtual Cells (AIVC)**—comprehensive computational models simulating cellular behavior across molecular, cellular, and tissue scales (Bunne et al 2024, Cell).

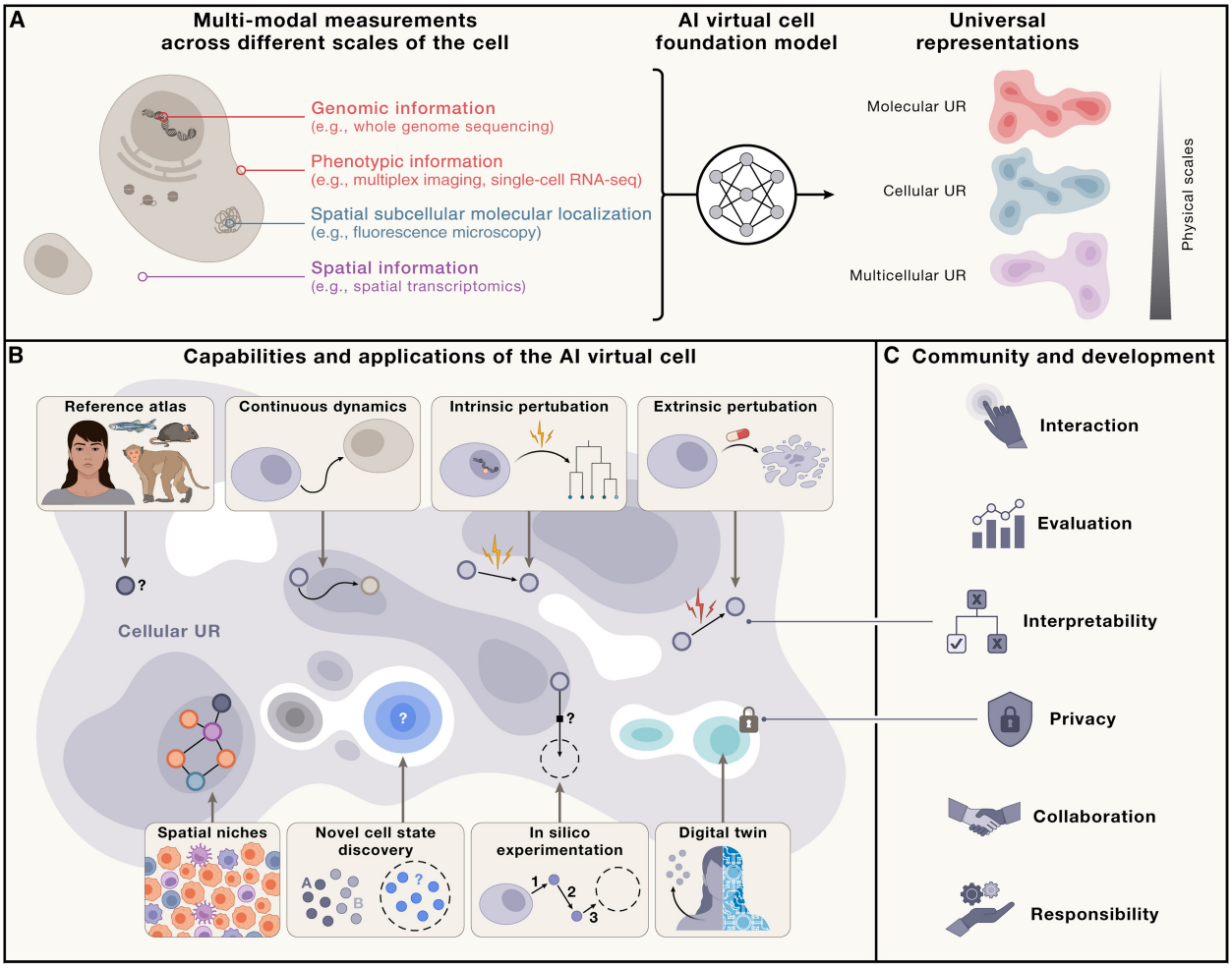

Figure 3: Bunne et al. 2024, Cell – Figure 1

Figure: Capabilities of the AI Virtual Cell. The AIVC provides universal representations (UR) of cell states

that can be obtained across species and conditions from different data modalities (A). These representations enable predicting cell biology, modeling dynamics, and performing in silico experiments (B). The utility depends on interactions at individual, community, and societal levels—requiring accessibility, interpretability, evaluation frameworks, privacy protection, and collaborative development (C). Source: Bunne et al 2024, Cell. License: CC-BY 4.0.

This enables **in silico experimentation**: 1. Simulate experiment computationally 2. Predict outcome with confidence intervals 3. Only test physically the most promising/uncertain predictions

Drug screening example:

Approach	Compounds	Cost	Time	Hits
Traditional	10,000 physical	$50M	2 years	5–10
AI-augmented	100M virtual → 1,000 physical	$5M	6 months	20–30

Active Learning: Scientists in the Loop

The most powerful approach combines AI prediction with human expertise:

```
Scientist's Question → Virtual Cell Simulation →
  Scientist Reviews + Domain Knowledge →
    Lab Experiments → Virtual Cell Learns → (Loop continues)
```

AI amplifies—doesn't replace—biological expertise. Scientists still ask questions, interpret meaning, decide what to test, and validate results. But now they can explore vastly larger hypothesis spaces.

What Biologists Need to Know About AI

Practical Skill Levels

Level	Who	What You Can Do	Time Investment
Consumer	All biologists	Use existing AI tools (AlphaFold, CADD scores)Interpret predictions criticallyUnderstand limitations and when to validateRecognize biases	Hours (this course)
User	Data-oriented	Run pre-trained models on your dataPerform data preprocessing and visualizationIntegrate AI into analysis pipelines	Weeks of practice
Developer	Computational biology	Fine-tune and train new modelsDevelop novel architecturesCollaborate as equal partner with ML researchers	Months to years

This textbook targets Levels 1–2.

When to Use AI

Use AI When:	Don't Use AI When:
Large datasets (1000+ examples)	Very little data (⟨100 examples)
Complex patterns (many variables)	Mechanism understanding is critical
Expensive/slow experiments	Very high stakes without validation
Need for scale (millions of predictions)	Problem is simple (basic statistics work)
Similar problems solved (transfer learning)	Training data doesn't match your population

Decision Framework:

```
Need to prioritize/predict many things?
  NO → Traditional experiments
  YES → Have >1000 training examples?
    NO → Use statistics or small ML models
    YES → Pattern too complex for simple rules?
      NO → Try simple models first (linear, random forest)
      YES → Consider deep learning
            ↓
      Always validate key predictions experimentally!
```

Common AI Failures and Lessons

Failure Type	Example	What Went Wrong	Lesson
Overfitting	Sepsis prediction: 80% accuracy in training, random chance in real hospitals	Learned when nurses check vitals, not sepsis biology	Validate on truly independent data from different sources
Unnecessary Complexity	Deep learning 85% vs. simple linear 87% accuracy	Problem was actually linear; complexity hurt performance	Start simple, only add complexity when needed
Population Bias	30% more variants flagged as "pathogenic" in African genomes	Training data ⟩80% European ancestry; novelty interpreted as pathogenicity	Ensure training data represents application population
Confounding	Gene X "causes" disease	Actually: Disease → Inflammation → Gene X	Draw causal models; design experiments to test

---

Summary

Key Takeaways:

1. **Biology's data explosion** outpaces experimental validation—AI helps prioritize what to test
2. **AI ⊃ ML ⊃ DL** – Hierarchy from broad to specific, each with distinct use cases
3. **Learning = optimization** – Algorithms discover patterns by adjusting parameters to minimize errors
4. **Correlation ≠ Causation** – AI finds associations; experiments establish causation through controlled perturbation

5. **Causal models over null hypotheses** – Compare competing biological mechanisms rather than testing "no effect"
6. **DAGs clarify causation** – Visual tools to distinguish direct effects from indirect correlations
7. **Proven impact** – AlphaFold, DeepVariant, single-cell analysis, drug discovery all transform research
8. **Paradigm shift** – From linear hypothesis-testing to iterative AI-augmented discovery loops
9. **Virtual cells and in silico experimentation** – AIVC enables computational simulation before physical testing
10. **Human expertise essential** – AI amplifies biological insight rather than replacing it
11. **Strategic application** – Use for large datasets, complex patterns, expensive experiments; avoid for small data, causal mechanisms, simple problems
12. **Critical evaluation needed** – Beware overfitting, bias, generalization failure, reproducibility issues

---

Key Terms

- **Artificial Intelligence (AI)**: Field of making computers perform tasks requiring human intelligence
- **Machine Learning (ML)**: Algorithms that learn patterns from data without explicit programming
- **Deep Learning (DL)**: ML using multi-layered neural networks
- **Causal Inference**: Methods for establishing causation, not just correlation
- **DAG (Directed Acyclic Graph)**: Visual representation of causal relationships where arrows indicate cause-and-effect
- **Parameters**: Numerical values a model learns to make predictions
- **Training Data**: Examples with known answers used to teach models
- **Optimization**: Adjusting parameters to minimize prediction errors
- **Overfitting**: Learning training data too well, failing on new data
- **Bias**: Systematic errors from unrepresentative training data
- **Confounding**: When a third variable creates spurious associations
- **Foundation Model**: Large-scale models trained on diverse data, transferable to many tasks
- **AI Virtual Cell (AIVC)**: Computational model simulating cellular behavior across scales
- **Active Learning**: Iterative process where AI identifies most informative next experiments
- **CRISPR knockout/knockdown**: Gene editing to delete or reduce gene expression for causal validation
- **Drug inhibition**: Chemical compounds blocking protein function to test causation
- **Overexpression**: Artificially increasing gene expression levels to observe phenotypic effects

---

Test Your Understanding: Can You Answer These?

1. What is the difference between AI, ML, and DL?

Answer:

- **AI (Artificial Intelligence)** is the broadest concept—any technique that enables computers to mimic human intelligence (e.g., BLAST for sequence alignment uses programmed rules)
- **ML (Machine Learning)** is a subset of AI where algorithms learn patterns from data without explicit programming (e.g., Random Forest classifier learns to predict variant pathogenicity from training examples)
- **DL (Deep Learning)** is a subset of ML that uses multi-layered neural networks to automatically discover hierarchical patterns (e.g., AlphaFold uses deep neural networks with many layers to predict protein structure)

Example: BLAST is AI but not ML (uses rules). A Random Forest variant classifier is ML but not DL (no neural networks). AlphaFold is DL (deep neural networks).

2. Why can't AI alone determine if a gene causes a disease?

Answer:

AI can only find **correlations** (patterns that occur together), not **causation** (one thing directly causing another).

Example: If Gene X expression correlates with Disease Y, there are multiple possible explanations: – Gene X directly causes Disease Y – Disease Y causes Gene X expression to change – A third factor (e.g., inflammation) causes both Gene X expression and Disease Y

AI cannot distinguish between these scenarios. Only **controlled experiments** (like CRISPR knockout, drug inhibition, or overexpression) can establish causation by directly perturbing the gene and observing if the disease phenotype changes.

3. When should you use AI versus simple experiments?

Answer:

Use AI when: – You have large datasets (thousands of samples) – You need to prioritize among many possibilities (e.g., which of 1 million variants to study?) – Experiments are expensive or time-consuming – You're looking for complex patterns humans might miss

Use simple experiments or statistics when: – You have small datasets (fewer than 100 samples) – You're testing a specific mechanistic hypothesis – The question is straightforward (e.g., comparing expression of 10 genes between two conditions) – You need to establish causation, not just correlation

Key principle: AI helps you decide what to test experimentally, but experiments prove why something happens.

Hands-On Labs

Lab 1.1: Getting Started with Google Colab and Python (30–45 min)

Learn: – Google Colab interface – Python basics for biologists – DNA/RNA sequence analysis – GC content, motif finding – Simple visualizations

Access Lab 1.1 on Google Colab

Lab 1.2: Essential Python Tools for Bioinformatics (60–90 min)

Learn: – NumPy for numerical computing – Pandas for organizing data – Matplotlib for publication plots – Biopython for sequences – Complete RNA-seq analysis workflow

Access Lab 1.2 on Google Colab

Chapter 2: From Biological Intuition to Deep Learning

You're in your room, and there's a small globe-shaped ball sitting on your desk—one of those foam stress balls shaped like Earth. Out of boredom between problem sets, you toss it in the air and catch it with your eyes closed. Your right index finger lands on··· **water**. You try again. **Water**. Again. **Land**. After 9 tosses, you've recorded: **W L W W W L W L W** (6 water, 3 land).

"So Earth is about 67% water?" you think.

But wait—are you certain? What if you just got lucky? What if the true proportion is actually 70%, and you happened to get 6 out of 9 by chance? Or maybe it's 60%? You decide to keep going. After 100 tosses, you have: **71 water, 29 land**. Now you're more confident: "Okay, probably around 71% water."

What just happened in your brain?

You started with complete uncertainty. Each toss gave you evidence. You continuously updated your belief about Earth's water proportion. By the 100th toss, you had a much sharper, more confident estimate.

This is Bayesian reasoning—and you do it every day without realizing it: – Troubleshooting failed PCRs – Deciding if a gene expression change is real or noise – Evaluating whether a called variant is real or a sequencing error – Interpreting your professor's claim that "this exam will be easy"

Your brain is already a sophisticated Bayesian inference machine. This chapter will show you that: 1. **You already think probabilistically** (you just didn't have the vocabulary) 2. **Biology itself operates on probability distributions** (not fixed values) 3. **Deep learning formalizes this intuition** at massive scale

By understanding this connection, neural networks will transform from mysterious black boxes into intuitive tools that amplify what you already do naturally.

Your Brain Already Does Bayesian Inference

You're a Natural Statistician

Every moment, your brain makes predictions and updates beliefs. Let's see this in action.

Example 1: "Mom Said I'd Lose Weight in College!" Before starting university (Prior belief):

Your mom confidently told you: "Don't worry about the tuition—you'll lose weight in college! Students are always busy, walking everywhere, no time to eat much."

```
Your initial belief:
• Probability of losing weight: 65%
• Reasoning: Busy schedule, independent living, more exercise
```

First semester reality (New evidence):

You're a student at Korea University. The actual data you observe:

```
Weekly schedule includes:
• Central Plaza (  ): Makgeolli (  , rice wine) gatherings 2x/week
• Daedongje (  , university festival): 3 days of continuous eating and drinking
• Goyeonjeon (  , Korea-Yonsei rivalry game): Epic celebration with traditional Korean food and drinks
• Late-night   (chimaek = chicken + beer): Every time you finish an exam
• Convenience store ramyeon: 1 AM study fuel
```

End of first semester (Posterior belief):

```
Actual outcome: Gained 7 kg

Updated beliefs:
• Probability of losing weight: 5% ← Dramatically decreased!
• Probability of gaining weight: 85% ← Reality hit hard
• Mom's prediction accuracy: 10% ← Love you mom, but...
```

What happened here?

You performed Bayesian inference: 1. **Prior:** Mom's belief that college = weight loss (65% confident) 2. **Evidence:** Actual data from Central Plaza, Daedongje, Goyeonjeon showed high-calorie social culture 3. **Likelihood:** This evidence strongly contradicts the prior belief 4. **Posterior:** Updated to 85% confident that college = weight GAIN

The math you didn't know you were doing:

```
Updated Belief (Posterior)   How well hypothesis explains evidence (Likelihood)
                              × What you believed before (Prior)

P(weight loss | college life data) = Very low!
P(weight gain | college life data) = Very high!
```

The evidence (막걸리□ at Central Plaza, festival food, late-night 치맥□□) completely overwhelmed your prior belief. This is Bayesian updating in action—when strong evidence contradicts your prior, you update your beliefs accordingly!□

Your brain did this automatically. You didn't consciously calculate probabilities, but you naturally weighted the evidence (all those makgeolli nights) against your prior belief (mom's prediction) and reached a new conclusion.

Example 2: "This Exam Will Be Easy" **Two weeks before the exam:** Your professor announces, "Don't worry everyone, this exam will be easy!"

Your automatic Bayesian reasoning:

```
Prior beliefs about exam difficulty:
Based on past exams from this professor:
• Last 3 exams: Average score was 68%, 71%, 65%
• Students always complain exams are hard
• Professor has said "easy" before... but wasn't

Initial belief: "Probably still going to be hard" → 70% confident it's hard

Evidence: Professor said "easy"
Likelihood: How often does "professor says easy" actually mean easy?
  Based on past experience: 20% of the time

Updated belief (Posterior):
"Still probably hard, but maybe slightly easier?" → 55% confident it's hard
```

After the exam: It was actually easy! Average score: 87%

Your brain updates again for next time:

```
New posterior becomes your next prior!

Next time professor says "easy":
Prior: 45% confident it's hard (down from 70%)
"Okay, maybe I should trust them more now"
```

This is Bayesian learning from experience! Each outcome updates your beliefs for future predictions.

Example 3: Is This Gene Differentially Expressed? You're analyzing RNA-seq data. Gene X shows: - Control: mean = 45.3 TPM (n=3 biological replicates) - Treatment: mean = 52.1 TPM (n=3 replicates) - p-value = 0.048

Your automatic Bayesian reasoning:

```
Question: "Is this change real or just noise?"

What you consider (even if unconsciously):
• Fold-change: 52.1/45.3 = 1.15 → small, modest evidence
• P-value: 0.048 → barely crosses 0.05 threshold
• Sample size: Only n=3 → high uncertainty
```

```
• Biology: Gene X is in a pathway known to respond to this treatment → increases plausibility
• Effect size: Absolute difference only 6.8 TPM → biologically meaningful?

Your conclusion: "Probably real, but I should validate with qRT-PCR"
                 → ~65% confidence it's a true change
```

Notice what you did: – Combined multiple pieces of evidence – Weighted each piece differently (biological knowledge matters!) – Didn't just rely on p-value – Expressed uncertainty (not "yes/no" but "probably")

This is Bayesian thinking!

You integrated: – **Prior:** Knowledge about the pathway (increases belief in real change) – **Likelihood:** Statistical evidence (fold-change, p-value, n) – **Posterior:** Updated belief incorporating everything (~65% confident)

Example 4: Is This Called Variant Real? You're reviewing whole-genome sequencing data. The variant caller identified a SNP at position chr7:55,242,466:

```
Reference: A
Called variant: G
Read support: 8 reads with G, 2 reads with A
Total depth: 10 reads
Base quality scores: mostly Q30+ (high confidence)
```

Your brain's automatic Bayesian analysis:

```
Starting belief (Prior):
• Most positions match reference: 99.9%
• True variant rate: ~0.1% per position

Evidence observed:
+ 8/10 reads support the variant
+ High base quality (Q30+)
- BUT: Only 10× depth (somewhat low)
- Position is in a repetitive region (increases error likelihood)

Two competing hypotheses:

H1: Real heterozygous variant (A/G genotype)
  Expected read distribution: ~50% A, ~50% G
  Observed: 20% A, 80% G → doesn't match perfectly, but sampling variation?

H2: Sequencing error (true genotype A/A)
  Expected: All A reads, but some errors
  Error rate in this region: ~2%
  Observed: 80% errors seems too high for random errors...

Updated belief (Posterior):
"Probably real, but confidence is moderate" → ~70% confident
"Should check in validation dataset or with different sequencing platform"
```

What happens with more evidence?

```
Additional evidence: You check gnomAD database
• This exact variant seen in 15/280,000 people (allele frequency ~ 0.00005)
• Found in multiple populations
• No evidence of being a common sequencing artifact
```

```
Updated posterior:
"Very likely real" → 95% confident

This is exactly what tools like DeepVariant do-but at genome-wide scale!
```

Key insight:

Variant calling is fundamentally a Bayesian problem: – **Prior:** Most positions match reference, true variants are rare – **Likelihood:** What's the probability of seeing this read pattern if it's real vs. error? – **Posterior:** Confidence that variant is real

Deep learning tools like DeepVariant learn to compute these posteriors automatically from millions of examples!

The Bayesian Formula (Don't Panic!)

You've been doing this intuitively. Here's the formal version:

```
Posterior  Likelihood × Prior

P(Hypothesis | Data)  P(Data | Hypothesis) × P(Hypothesis)
      ↑                       ↑                    ↑
  What you believe        How well this        What you believed
  AFTER seeing data     hypothesis explains     BEFORE seeing data
                         what you observed
```

In plain English:

> **Your updated belief about something = (How well it explains what you observed) × (What you thought before observing)**

Real examples from your life:

Situation	Prior	Likelihood	Posterior
College weight	Mom says you'll lose weight (65%)	Central Plaza 막걸리, 대동제, 고연전 치맥	Actually gained weight (85%)
Professor says "easy exam"	Past exams were hard (70%)	Prof said "easy" but was wrong before	Still probably hard (55%)
Gene expression	Pathway expected to respond (60%)	Small fold-change, p=0.048	Moderately confident (65%)
Called variant	Most positions match reference (99.9%)	8/10 reads + seen in gnomAD	Likely real (95%)

The beautiful part: Your brain weights evidence automatically. Strong evidence (makgeolli nights at Central Plaza, variant in gnomAD database) can overwhelm an optimistic or pessimistic prior. Weak evidence (professor says "easy") barely moves your belief.

Biology Is Fundamentally Probabilistic

Here's a profound insight that will change how you think about biology. Biology doesn't operate with fixed values. It operates with probability distributions.

What Does This Mean?

Common misconception: "Gene X is expressed at 47.3 TPM"

Reality: Gene X's expression is a probability distribution across cells, time, and conditions.

Let's see why.

Example 1: Gene Expression is a Distribution, Not a Number

Imagine you do single-cell RNA-seq on "identical" T cells from the same person, same tissue, same moment. You measure IL2 (interleukin-2) expression:

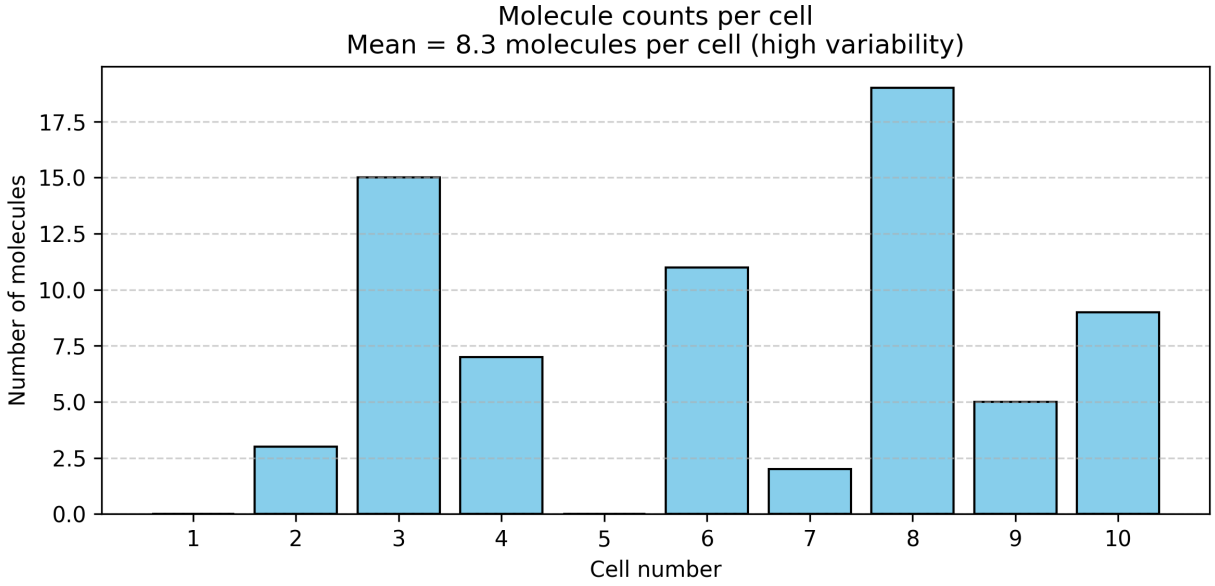

Figure 4: Molecule counts per cell

Figure: IL2 expression varies dramatically across individual cells. Single-cell measurements of IL2 (interleukin-2) mRNA molecules in 10 "identical" T cells from the same sample. Despite being genetically identical cells in the same condition, expression ranges from 2 to 19 molecules per cell, with a mean of 8.3. This huge variability (coefficient of variation ~70%) is not measurement error—it's biological reality driven by stochastic transcription.

Why does this happen?

Gene expression is fundamentally **stochastic** (random): - Transcription factors bind and unbind randomly - RNA polymerase initiation is probabilistic—sometimes it starts, sometimes it doesn't - mRNA molecules degrade at random times - Cells are in different cell cycle phases - Local microenvironment varies slightly between cells

This variability isn't measurement error—it's biological reality!

The "true" expression of IL2 isn't a single number. It's a **probability distribution**:

Figure: IL2 expression as a probability distribution. The same data shown as a probability distribution rather than individual cell measurements. Most cells (31%) express 1–5 molecules, but there's substantial spread: 12% express nothing, while 10% express more than 15 molecules. This distribution shape—not just the mean—is the biological reality.

Implications:

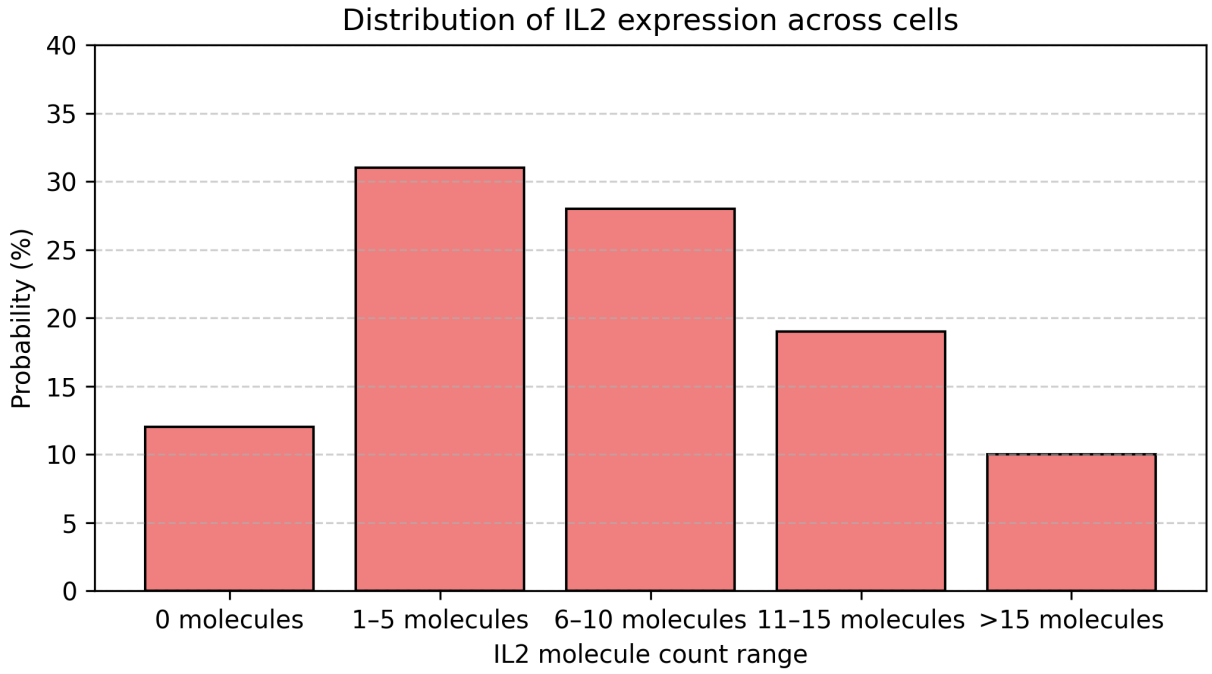

Figure 5: Distribution of IL2 expression across cells

When we say "IL2 is expressed at 8.3 TPM," we're really reporting the **center of a distribution**, not a fixed truth. The full story requires the entire distribution—mean, variance, shape.

This matters for AI/ML: Deep learning models should predict distributions, not just point estimates!

Example 2: Protein Folding is an Ensemble, Not a Structure

What you learned in biochemistry: "Protein X folds into structure Y"

What actually happens: Protein X samples a statistical ensemble of conformations:

Figure: Proteins exist as an ensemble of conformational states. Rather than adopting a single rigid structure, proteins constantly fluctuate between different conformations. The native state (lowest energy) is most probable (75%), but the protein regularly visits alternative conformations (20%), partially unfolded states (4%), and rarely completely unfolds (1%). This distribution changes with temperature, pH, mutations, and binding partners.

The protein constantly fluctuates! It: – Spends most time in the native state (most stable) – Occasionally visits alternative conformations – Rarely unfolds completely – The distribution changes with temperature, pH, mutations, binding partners

AlphaFold doesn't predict "the" structure—it predicts the **most probable** structure (the peak of the distribution). The pLDDT score tells you how confident AlphaFold is, which relates to how sharp that distribution is.

```
High pLDDT (>90): Sharp peak → protein is rigid, well-defined structure
Low pLDDT (<50): Flat distribution → protein is flexible, intrinsically disordered
```

Example 3: Evolution is Bayesian Updating Over Generations

Evolution literally performs Bayesian inference at the population level!

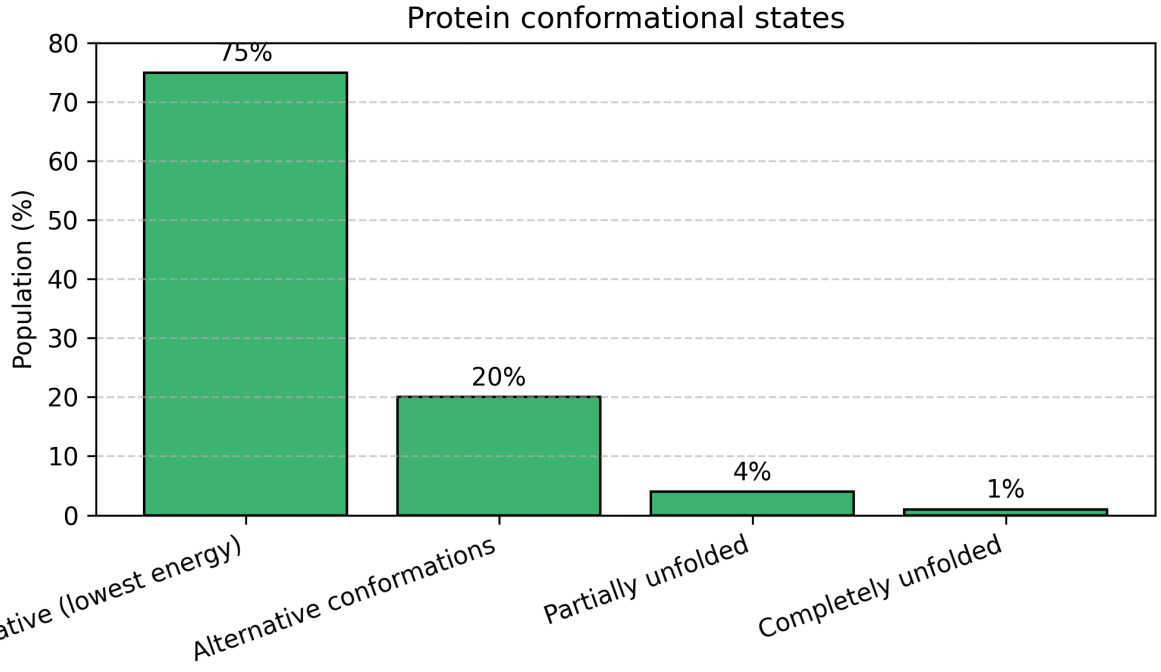

Figure 6: Protein conformational states

Before environmental change (Prior distribution):

Figure: Giraffe neck length distribution before environmental change. In the ancestral giraffe population, most individuals (80%) had short necks (⟨2m), with progressively fewer having medium (15%) or long (5%) necks. This represents the "prior" distribution before selection pressure.

Environmental change occurs: Climate shifts, tall trees become dominant food source.

Survival rates (Likelihood):

```
P(survive to reproduce | short neck) = 8%   ← Can't reach food
P(survive to reproduce | medium neck) = 35% ← Can reach some food
P(survive to reproduce | long neck) = 72%  ← Can reach most food
```

After several generations (Posterior distribution):

Figure: Giraffe neck length distribution after natural selection. After multiple generations of selection for tall tree feeding, the distribution has dramatically shifted. Long necks increased from 5% to 60% (12× increase!), medium necks doubled from 15% to 28%, while short necks decreased from 80% to 12%. This is Bayesian updating in action: the population "learned" which trait is adaptive based on survival data.

This is Bayesian updating!

```
P(long neck | survival)   P(survival | long neck) × P(long neck)
       ↑                          ↑                      ↑
  New frequency              Fitness advantage      Old frequency
   (Posterior)                 (Likelihood)          (Prior)
```

Evolution updates the population distribution based on environmental "evidence"! Natural selection is nature's way of performing Bayesian inference. Populations "learn" which traits are adaptive through survival data.

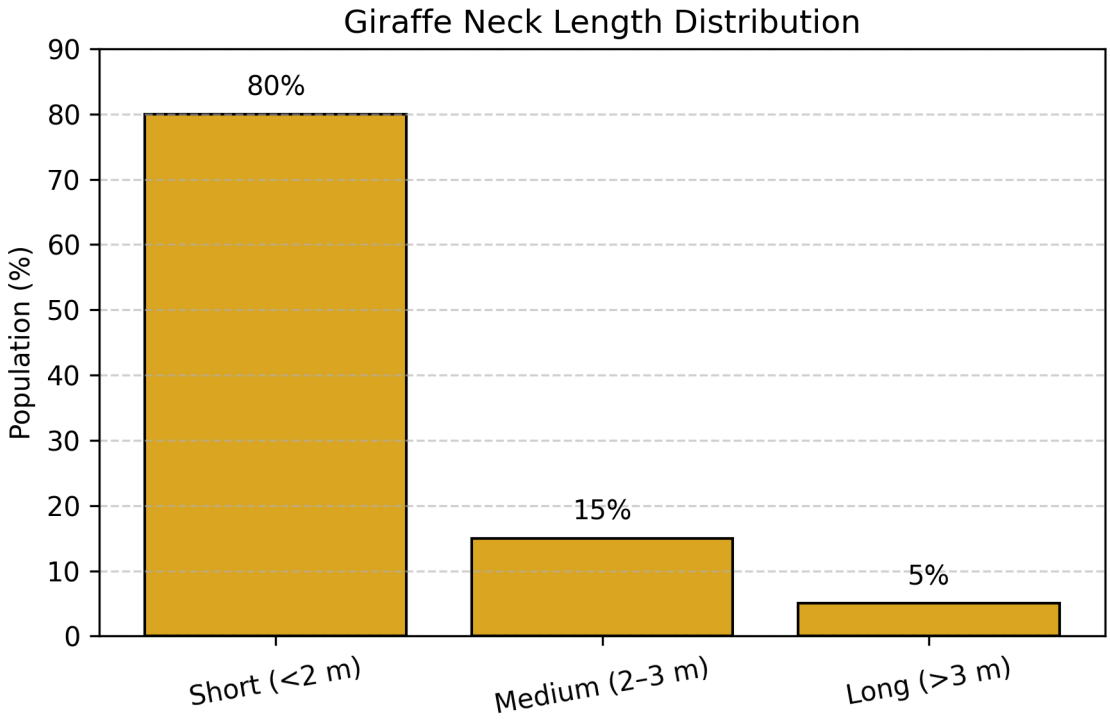

Figure 7: Giraffe Neck Length Distribution

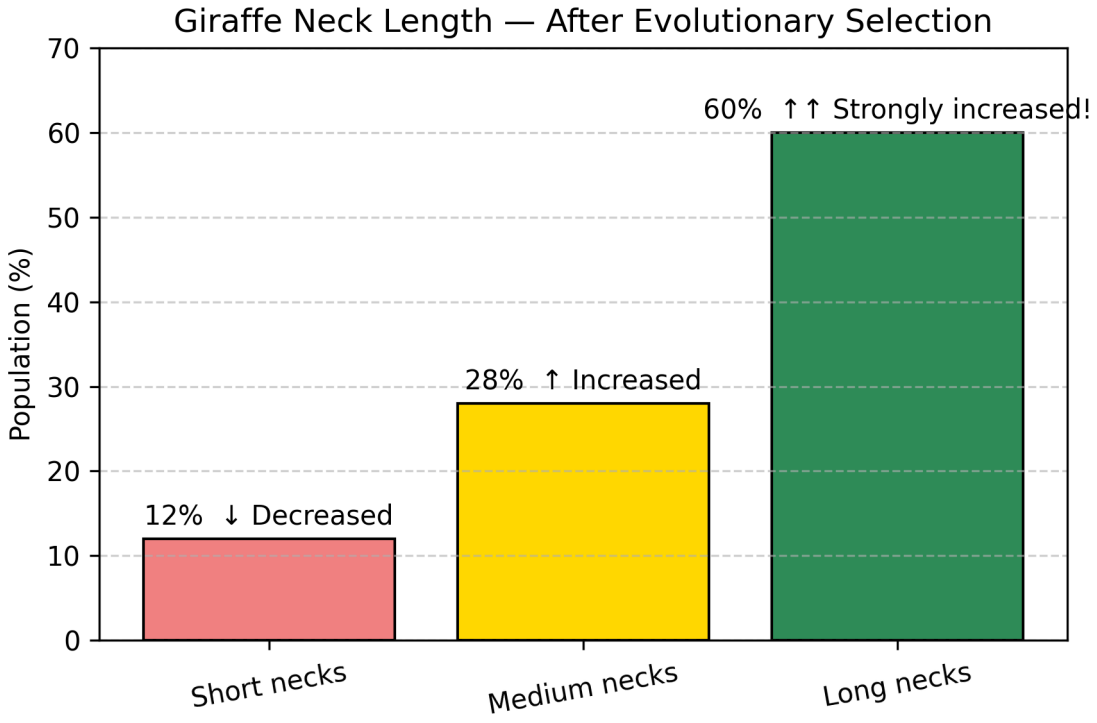

Figure 8: Giraffe Neck Length — After Evolutionary Selection

Example 4: Your Immune System Learns Like a Bayesian

Your immune system is a beautiful example of Bayesian learning in action.

First exposure to pathogen (Prior):

```
Your immune system has never seen this virus.

Antibody repertoire: Millions of random antibodies
Success rate: ~0.01% of antibodies can bind this pathogen
```

During infection (Observing data):

```
B cells produce antibodies and test them:
• Antibody A tries to bind virus → FAIL → B cell dies
• Antibody B tries to bind virus → FAIL → B cell dies
• Antibody C tries to bind virus → SUCCESS! → B cell survives & multiplies
• Antibody D tries to bind virus → SUCCESS! → B cell survives & multiplies
...

Evidence accumulates: "These specific antibodies work against the pathogen!"
```

After infection (Posterior = Immunological memory):

```
Memory B cells: Enriched for successful antibodies
Success rate: ~80% of memory B cells produce effective antibodies

Your immune system "learned" from data!
Improvement: 0.01% → 80% (8,000× better!)
```

Upon re-infection:

```
Prior = Posterior from last time
→ Rapid, effective response because you "remember" which antibodies work

This is why vaccines work!
```

The Bayesian interpretation:

1. **Prior:** Broad distribution of random antibodies (uniform prior)
2. **Likelihood:** Which antibodies successfully bind pathogen (data)
3. **Posterior:** Enriched distribution favoring effective antibodies (learned)
4. **Next infection:** Start with this posterior as new prior (memory)

The Pattern: Biology = Probability Distributions

Notice the theme?

Phenomenon	Not "The Value"	But "The Distribution"
Gene expression	"Gene X = 50 TPM"	P(expression | cell, state, time)
Protein structure	"Folds to structure S"	P(conformation | sequence, environment)
Evolution	"Giraffes have long necks"	P(trait | environment, time)
Immune response	"Antibody Y fights virus Z"	P(antibody effective | pathogen, history)
Genetic penetrance	"Mutation causes disease"	P(disease | genotype, background, environment)
Transcription	"Gene is ON"	P(transcription event | TF levels, chromatin state)

Profound implication: When we do biology, we're not discovering fixed truths—we're characterizing probability distributions over biological states!

Why this matters for AI:

Modern AI tools should: – Output distributions, not just point estimates – Quantify uncertainty – Update beliefs as new evidence arrives

Understanding this probabilistic foundation helps you use and interpret AI tools correctly!

The Globe–Tossing Experiment: Learning from Data

Now let's formalize these intuitions with a beautiful example from Richard McElreath's Statistical Rethinking.

The Setup

Back to your room with that globe–shaped ball.

You've never actually checked what proportion of Earth is water. Time to find out!

Your method: 1. Toss the ball in the air 2. Catch it with your eyes closed 3. Note what's under your right index finger: **Water (W)** or **Land (L)** 4. Repeat many times

First 9 tosses: W L W W W L W L W

Data summary: 6 Water, 3 Land

Question: What proportion of Earth is water?

The Naïve Answer

"I got water 6 out of 9 times, so Earth must be 6/9 = 67% water!"

But wait—are you certain?

- What if the true proportion is 70%, and you just happened to get 6 W's by chance?
- What if it's 60%, and you got slightly lucky?
- With only 9 tosses, there's substantial uncertainty!

Bayesian approach: Express belief as a **probability distribution** over all possible proportions, not a single number!

Considering Different Hypotheses

Let's evaluate several candidate hypotheses:

```
H : p = 0.00 (0% water - all land)
H : p = 0.25 (25% water)
H : p = 0.50 (50% water)
H : p = 0.75 (75% water)
H : p = 1.00 (100% water - all ocean)
```

Before Any Data: The Prior

You've never checked before. You have no idea what to expect.

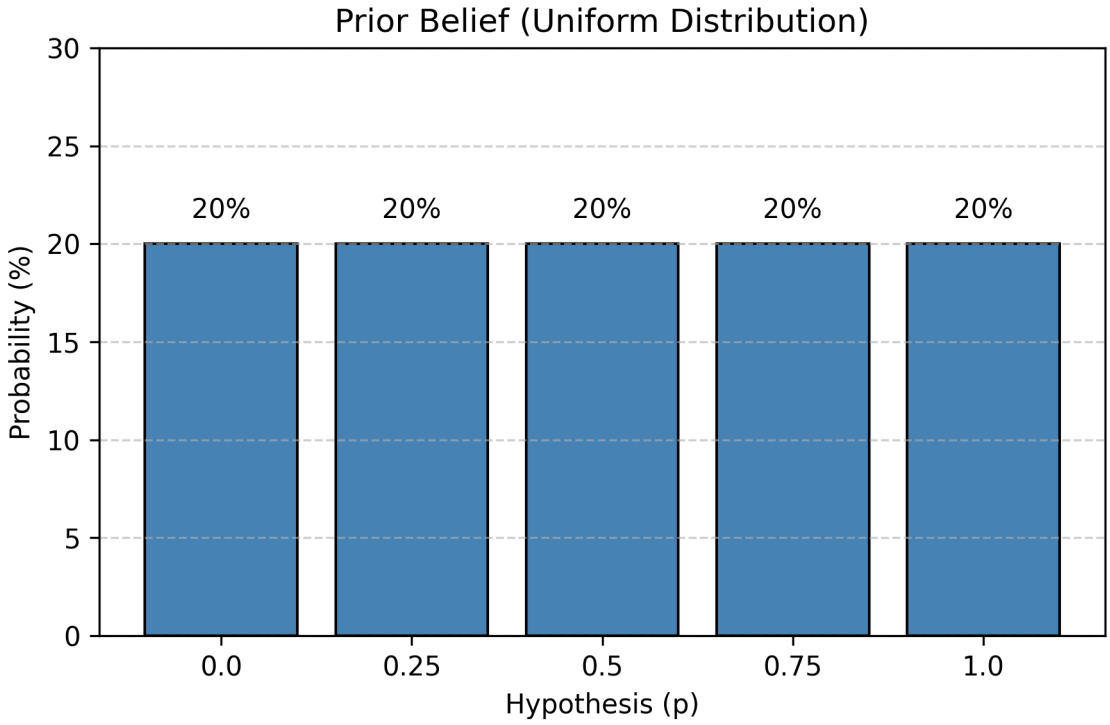

Figure 9: Prior Belief (Uniform Distribution)

Figure: Uniform prior distribution. Before observing any data, all hypotheses are equally plausible. Each of the five water proportions (0%, 25%, 50%, 75%, 100%) has equal 20% probability. This represents complete uncertainty—we have no reason to favor one hypothesis over another.

This is called a **uniform prior**—no hypothesis is favored initially.

After Seeing Data: The Likelihood

Now comes the clever part. We ask:

"For each hypothesis, how likely would we observe 6 W and 3 L?"

This is the **likelihood**: P(data | hypothesis)

Understanding Likelihood Through the "Garden of Forking Paths" This is Richard McElreath's brilliant intuitive explanation!

Imagine each hypothesis as a bag of marbles: – **Blue marbles** = water – **White marbles** = land

For p = 0.50 (bag has 1 blue, 1 white marble):

```
Toss 1: W
  Ways to get W: 1 (pick the blue marble)
  Paths so far: 1

Toss 2: L
  For each previous path, ways to get L: 1 (pick white marble)
  Paths so far: 1 × 1 = 1
```

```
Toss 3: W
  Ways to get W: 1
  Paths so far: 1 × 1 = 1

Continue for all 9 tosses (6 W's and 3 L's)...

Total paths: 1^6 × 1^3 = 1
```

For p = 0.75 (bag has 3 blue, 1 white marble):

```
Each W: 3 ways (pick any of 3 blue marbles)
Each L: 1 way (pick the 1 white marble)

Total paths: 3^6 × 1^3 = 729 paths!
```

For p = 0.25 (bag has 1 blue, 3 white marbles):

```
Each W: 1 way
Each L: 3 ways

Total paths: 1^6 × 3^3 = 27 paths
```

For p = 0.00 or p = 1.00:

```
Can't produce both W and L from these bags
Paths = 0
```

The key insight:

Likelihood = Number of ways this hypothesis could produce your observed data

More paths = more plausible!

Calculating the Posterior

Now we combine prior and likelihood using the Bayesian formula:

Figure: Final posterior probabilities after 9 tosses. After multiplying likelihood (paths) by prior and normalizing, we get the posterior distribution. The data (6W, 3L) provides overwhelming evidence for p=0.75, which receives 97.6% of the probability mass. The hypothesis p=0.25 gets only 2.3%, and p=0.50 is nearly ruled out at 0.1%. The extreme values (p=0 and p=1.0) are impossible since we observed both water and land.

Result: After seeing 6 W and 3 L: – 97.6% confidence that p □ 0.75 – 2.3% confidence that p □ 0.25 – Other values essentially ruled out

Visualizing Bayesian Updating

Let's see how beliefs evolve with each toss:

Figure: Sequential Bayesian updating with each observation. Starting from a flat uniform prior (blue line), each observation shifts and sharpens the distribution. After the first water (W, orange line), belief shifts toward higher p values. After one water and one land (W L, green line), the distribution centers around p=0.5. After all 9 tosses with 6 waters and 3 lands (red line), the distribution forms a sharp peak around p=0.67. Notice how each piece of evidence updates the previous belief, and more data leads to stronger confidence (narrower distribution).

Key observations:

1. **Each toss updates beliefs** – the distribution shifts and sharpens
2. **More data = more confidence** – the peak gets narrower

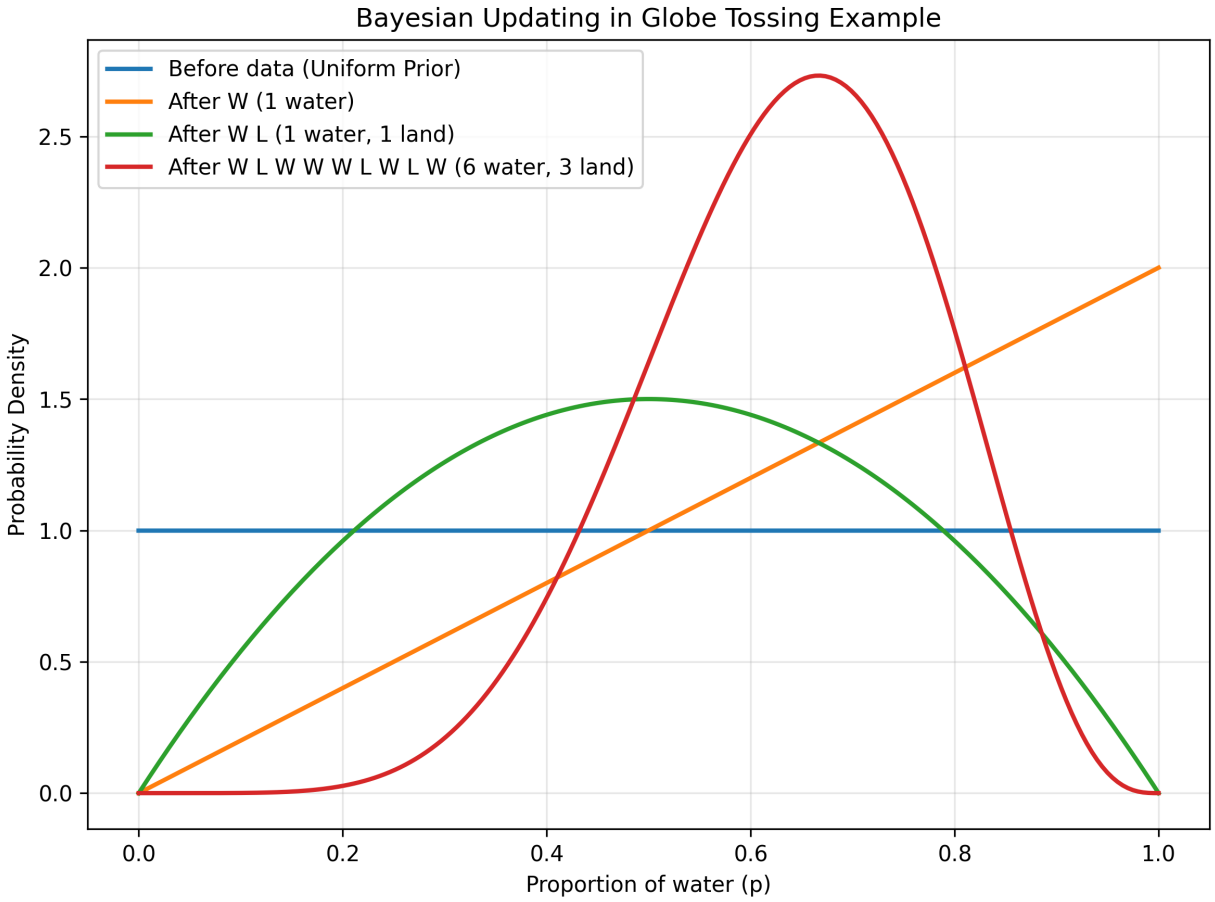

Figure 10: Posterior ∝ Likelihood × Prior (After Normalization)

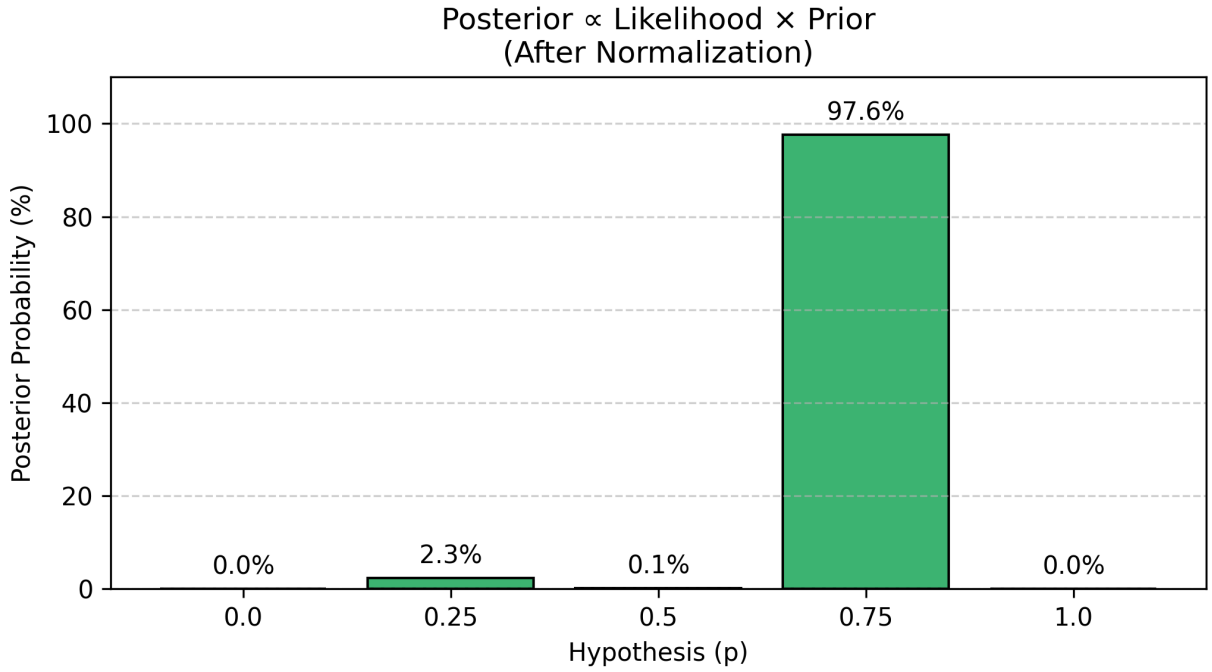

Figure 11: Bayesian Updating in Globe Tossing Example

3. **Evidence accumulates** – early random fluctuations smooth out
4. **Never 100% certain** – there's always some uncertainty (the distribution has width)

The Power of More Data

Let's continue tossing!

Figure: Posterior distribution sharpens dramatically with more data. After 100 tosses (71W, 29L), the posterior distribution becomes much narrower and taller. The MAP estimate shifts slightly to 0.71, and the 95% confidence interval shrinks from (0.45–0.85) with 9 tosses to (0.62–0.79) with 100 tosses. This demonstrates a fundamental principle: more data reduces uncertainty, giving us sharper, more confident predictions.

With 100 tosses: – Best estimate: p □ 0.71 (71%) – 95% confidence interval: 0.62 – 0.79 – Much sharper than with 9 tosses!

More data doesn't just change your point estimate—it **reduces uncertainty**. The posterior distribution becomes tighter, narrower, more confident.

The Bayesian Formulation

For those wanting the mathematical details:

```
For each possible proportion p, calculate:

Posterior(p)  Likelihood(data | p) × Prior(p)

Where Likelihood is the binomial probability:
L(p | k successes in n trials) = C(n,k) × p^k × (1-p)^(n-k)
```

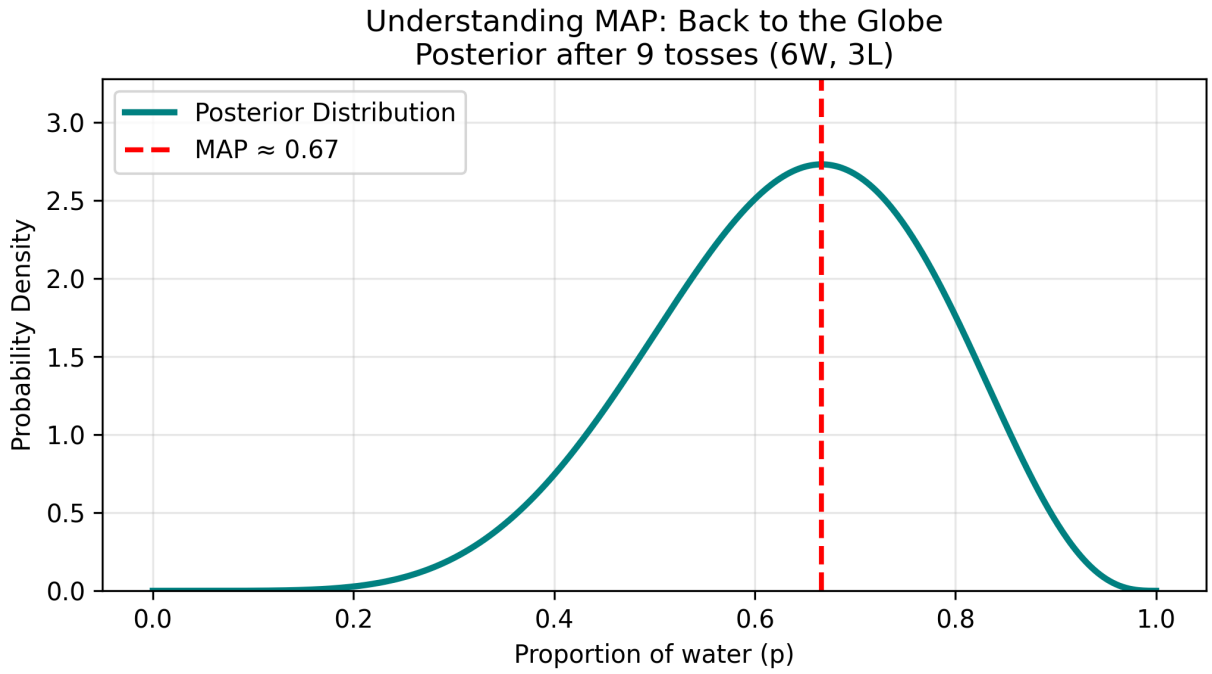

Figure 12: The Power of More Data – Posterior after 100 tosses

```
C(n,k) = "n choose k" = n! / (k!(n-k)!)
```

For our data (6 W in 9 tosses):

```
L(p=0.75) = C(9,6) × 0.75^6 × 0.25^3
          = 84 × 0.178 × 0.0156
          = 0.234

L(p=0.67) = C(9,6) × 0.67^6 × 0.33^3
          = 84 × 0.090 × 0.036
          = 0.272 (highest!)

L(p=0.50) = C(9,6) × 0.50^6 × 0.50^3
          = 84 × 0.0156 × 0.125
          = 0.164
```

The peak of the likelihood is at p = 6/9 = 0.67, which is the maximum likelihood estimate (MLE). But Bayesian inference gives us the entire distribution, not just the peak!

From Bayesian Inference to Deep Learning

Now for the crucial connection: **How does deep learning relate to all this Bayesian reasoning?**

The Ideal vs. The Practical

Ideal Bayesian inference:

```
For EVERY possible hypothesis, calculate:
```

```
Posterior(hypothesis | data) = [Likelihood × Prior] / Evidence

Keep the ENTIRE distribution!
```

Problem: For complex problems, this is computationally impossible.

Example: AlphaFold predicting protein structure

- Input: 300 amino acid sequence
- Output: 3D coordinates for ~5,000 atoms
- Possible conformations: ~10^300 (more than atoms in the universe!)
- Computing exact posterior over all conformations: Would take trillions of years

What do we do?

We make a practical compromise:

Deep Learning solution:

"Let's find the SINGLE BEST hypothesis (maximum posterior) really quickly!"

```
Bayesian (ideal):
"Structure 1: 28%, Structure 2: 19%, Structure 3: 15%, Structure 4: 12%, ..."
→ Keep full probability distribution over all structures

Deep Learning (practical):
"Structure 1 is most likely!"
→ Find the single best answer fast
```

This approximation is called **Maximum A Posteriori (MAP) estimation**.

Understanding MAP: Back to the Globe

Let's make this concrete with our globe-tossing example.

Figure: Posterior distribution and MAP estimate after 9 tosses. The posterior distribution (teal curve) shows our beliefs about the water proportion after observing 6 waters and 3 lands. The distribution peaks around p=0.67, which is the Maximum A Posteriori (MAP) estimate—the single most likely value. However, the distribution has substantial width, indicating uncertainty: plausible values range from roughly 0.45 to 0.85.

Two ways to use this distribution:

Full Bayesian approach:

```
"I believe Earth is:
 - 75% water with 40% confidence
 - 70% water with 35% confidence
 - 67% water with 25% confidence
 - 80% water with 15% confidence
 ..."

Keep the ENTIRE distribution in memory!
```

MAP approach (what deep learning does):

```
"Earth is most likely 75% water."
→ Just report the PEAK of the distribution
→ Ignore the rest
```

Why MAP?

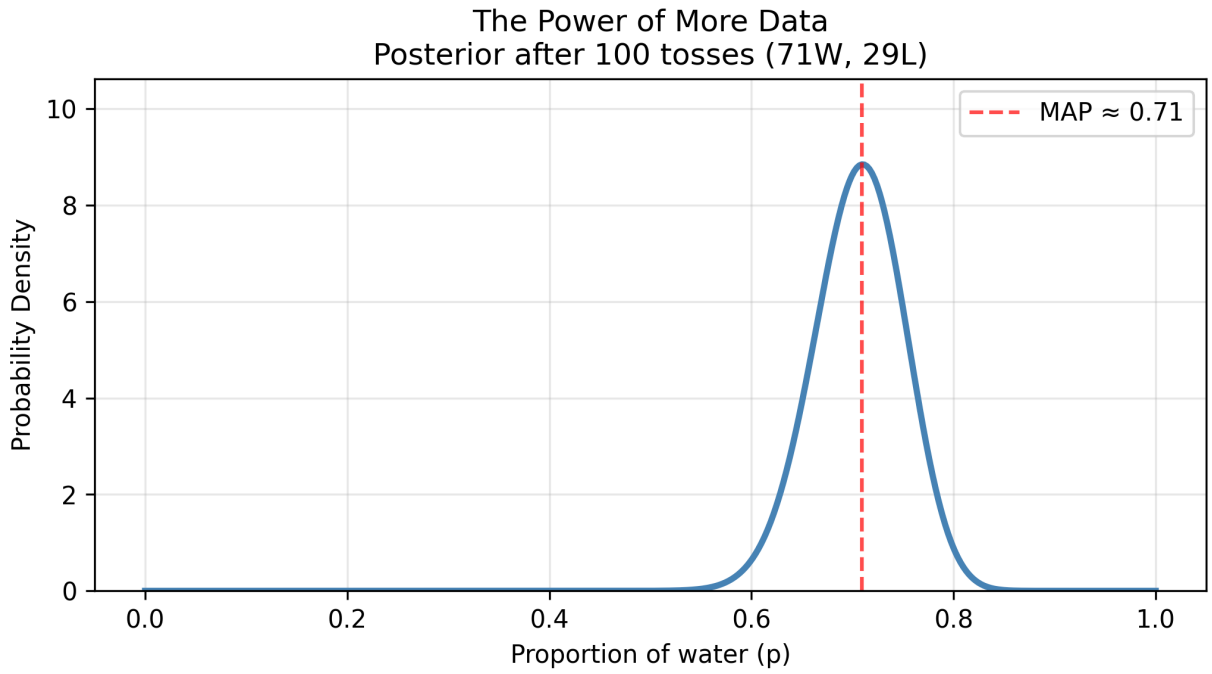

Figure 13: Understanding MAP: Back to the Globe – Posterior after 9 tosses

It's **much faster** and uses **much less memory**. For the globe example, the difference is small. But for AlphaFold predicting protein structures:

```
Full Bayesian:
Store probability for 10^300 possible structures
→ Impossible!

MAP:
Find the single most probable structure
→ Doable in hours!
```

The Mathematical Connection: Loss Functions

Now let's understand **how** deep learning finds that peak (the MAP estimate).

What is a Loss Function? Think back to the globe example. For each hypothesis (p = 0.25, 0.50, 0.75, etc.), we calculated how many "paths" could produce our data:

```
Hypothesis | Paths (Likelihood) | Which means...
-----------|--------------------|---------------
p = 0.25   | 27 paths           | Somewhat plausible
p = 0.50   | 1 path             | Very implausible!
p = 0.75   | 729 paths          | Very plausible!
```

We want to find the hypothesis with the MOST paths (highest likelihood).

Deep learning does this by turning it into an optimization problem:

```
Instead of: "Find hypothesis with most paths"
```

```
We say: "Find hypothesis that minimizes 'badness'"

Where "badness" = How unlikely the data is under this hypothesis
```

This "badness" is called the **loss function**.

Loss Function = "How Wrong Am I?" Let's make this concrete with variant calling.

Scenario: You're training a neural network to predict if variants are real or sequencing errors.

Training example: – Variant features: 8/10 reads support alternate allele, high base quality – True label: Real variant (y = 1)

Your neural network's prediction: – Probability it's real: 0.95 ($\hat{y} = 0.95$)

How do we measure if this prediction is good?

```
Loss = "How surprised should I be if the true label is y,
        but my model predicted ŷ?"

If y = 1 (real variant):
  Prediction ŷ = 0.95 → Loss = 0.05 (low! Good prediction!)
  Prediction ŷ = 0.50 → Loss = 0.69 (medium, uncertain)
  Prediction ŷ = 0.10 → Loss = 2.30 (high! Bad prediction!)
```

The mathematical trick:

```
Loss = -log(probability of truth)

If model says 95% confident and is correct:
Loss = -log(0.95) = 0.05

If model says 10% confident but truth is opposite:
Loss = -log(0.10) = 2.30
```

Why negative log?

```
High probability (0.9) → Small negative log (0.1) → Small loss
Low probability (0.1) → Large negative log (2.3) → Large loss

It converts "maximize probability" into "minimize loss"
(Computers are better at minimizing than maximizing!)
```

Connecting to Bayesian Likelihood Remember the Bayesian formula?

```
Posterior  Likelihood × Prior

We want to maximize: P(data | hypothesis)
                        ↑
                  This is likelihood!
```

In deep learning:

```
Training objective: Minimize loss across all training examples

Loss = -log P(data | parameters)
       ↑
   This is negative log-likelihood!

Minimizing loss = Maximizing likelihood!
```

Concrete example:

You have 3 training variants:

```
Variant 1: True label = Real (1), Model predicts 0.9
  Loss  = -log(0.9) = 0.105

Variant 2: True label = Error (0), Model predicts 0.2
  Loss  = -log(0.8) = 0.223  [note: 0.2 for "real" means 0.8 for "error"]

Variant 3: True label = Real (1), Model predicts 0.7
  Loss  = -log(0.7) = 0.357

Total Loss = 0.105 + 0.223 + 0.357 = 0.685
```

Goal of training: Adjust model parameters (weights) to minimize this total loss!

When loss is minimized → Model's predictions match the data well → We found the maximum likelihood!

Training = Climbing to the Peak

Now we understand: – **Loss function** = How badly the model explains the data – **Minimizing loss** = Maximizing how well the model explains the data – This is **exactly** what Bayesian inference does with likelihood!

The training process:

```
Step 1: Start with random parameters (random hypothesis)
        Loss is high (model explains data poorly)
        Starting point: gray dot at low posterior

Step 2: Adjust parameters slightly
        Calculate new loss
        Did loss decrease? Good! Keep this change.
        Did loss increase? Bad! Try different direction.
        Moving upward: pink dot

Step 3: Continue adjusting parameters
        Loss keeps decreasing
        Model explains data better
        Climbing higher: yellow dot

Step 4: Keep moving toward the peak
        Posterior probability increasing
        Getting closer: yellow dot near top

Step 5: Reach the maximum!
        Found the peak! (MAP estimate)
        Green dot at the highest point
        Stop when loss stops decreasing
```

Visualizing this:

Figure: Gradient descent climbs to the maximum posterior. Training a neural network is like climbing a hill to find the peak. Starting from a random position (Step 1, gray dot at bottom left with low posterior probability), the algorithm iteratively adjusts parameters to move uphill. At each step (Steps 2–5, marked by colored dots), the loss decreases (equivalently, posterior probability increases). The red dashed line marks the Maximum A Posteriori (MAP) estimate—the highest point on the posterior distribution. Step

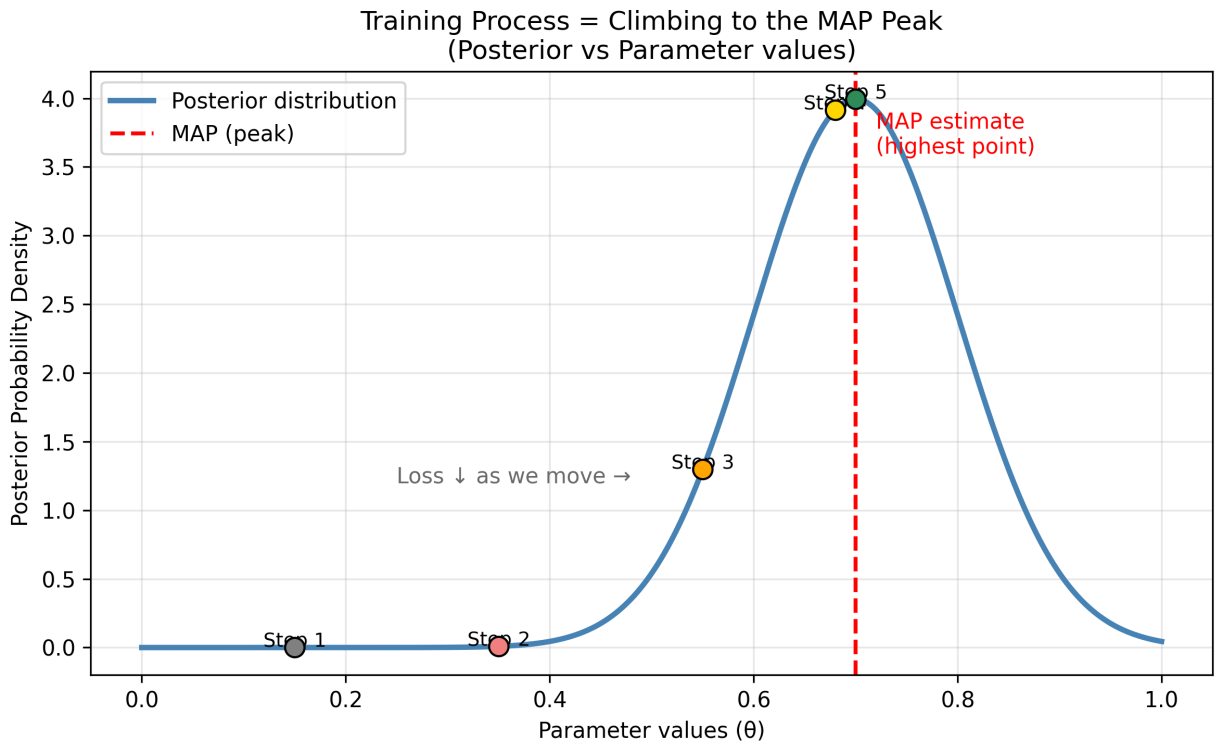

Figure 14: Training Process = Climbing to the MAP Peak

5 (green dot) reaches the peak, where the model's parameters best explain the observed data. This is gradient ascent on the posterior landscape (or equivalently, gradient descent on the loss landscape).

This process is called **gradient descent**—we "descend" down the loss landscape (or equivalently, "ascend" up the posterior landscape) to find the peak.

One More Thing: Preferring Simpler Models

Remember in the globe example, we could use different priors?

```
Uniform prior: All hypotheses equally likely initially
Informative prior: Some hypotheses more likely before seeing data
```

Deep learning has something similar: **preferring simpler models**.

The idea:

```
Imagine two models both explain the data perfectly:

Model A: Uses weights [0.3, -0.2, 0.1]
         Simple, small numbers

Model B: Uses weights [25.7, -31.4, 18.9]
         Complex, large numbers

Which do you trust more?
```

Most scientists (and mathematicians) prefer Model A because of **Occam's Razor**: "Simpler explanations are usually better."

In deep learning, we implement this preference:

```
Modified Loss = How wrong the model is + Penalty for complexity

Total Loss = Original Loss + (small number) × (sum of weight magnitudes)
                  ↓                              ↓
            Likelihood term              "Prior" that prefers simplicity
```

Example:

```
Model A: Loss = 0.50, Weights = [0.3, -0.2, 0.1]
  Total = 0.50 + 0.01 × (0.3 + 0.2 + 0.1) = 0.50 + 0.006 = 0.506

Model B: Loss = 0.45, Weights = [25.7, -31.4, 18.9]
  Total = 0.45 + 0.01 × (25.7 + 31.4 + 18.9) = 0.45 + 0.76 = 1.21

Even though Model B fits data slightly better (0.45 < 0.50),
Model A wins because it's much simpler!
Total: 0.506 < 1.21
```

This complexity penalty is like having a **prior belief** that weights should be small!

```
Bayesian: Prior = "I believe simpler hypotheses are more likely"
Deep Learning: Regularization = "Penalize large weights"

Same idea, different implementation!
```

Summary: The Bayesian–Deep Learning Dictionary

Bayesian Inference	Deep Learning	Intuition
Likelihood P(data\	hypothesis)	Loss function
Maximize likelihood	Minimize loss	Same goal, flipped!
Prior P(hypothesis)	Regularization penalty	“Prefer simpler explanations”
Posterior P(hypothesis\	data)	Trained model
Full probability distribution	Single best parameters (MAP)	Trade full info for speed
Bayesian updating	Training (gradient descent)	Improve guess iteratively

Deep learning is Bayesian inference with a practical twist: 1. **Approximation:** Find single best answer (MAP) instead of full distribution 2. **Optimization:** Turn “maximize posterior” into “minimize loss” 3. **Scale:** Handle millions of parameters efficiently with GPUs 4. **Speed:** Compute in seconds/hours instead of years 5. **Flexibility:** Learn complex non-linear patterns through neural networks

Why Not Keep the Full Distribution?

Why do we settle for MAP (single best answer) instead of full Bayesian inference?

Computational cost:

```
Full Bayesian:
"Variant pathogenicity could be:
 65% with 30% confidence,
 70% with 40% confidence,
 60% with 20% confidence, ..."
```

```
→ Store and compute with entire distribution
→ Very slow for complex models

Deep Learning (MAP):
"Variant is 67% likely pathogenic"
→ Single number
→ 1000× faster
```

For real problems: – Full Bayesian deep learning: Days to weeks of computation – Standard deep learning: Minutes to hours – Difference: 100–1000× speedup

When full Bayesian deep learning is used:

Some applications DO keep uncertainty estimates:

```python
# Bayesian Neural Network
predictions = model.predict(variant, num_samples=100)
mean_pred = predictions.mean()
uncertainty = predictions.std()

print(f"Pathogenic: {mean_pred:.1%} ± {uncertainty:.1%}")
# "This variant is 73% ± 12% likely pathogenic"
```

This is valuable when: – Medical diagnosis (need to know uncertainty!) – Autonomous vehicles (safety critical) – Drug discovery (expensive to test, want confidence) – Scientific discovery (want to know what we don't know)

Recent research: Bayesian deep learning is an active area! Methods like: – Monte Carlo Dropout – Variational inference – Ensemble methods

…provide uncertainty estimates at reasonable computational cost.

Why This Connection Matters for Genomics

Understanding the Bayesian foundation of deep learning helps you use AI tools more effectively and critically.

Interpret Model Confidence Correctly

Scenario: A variant predictor outputs "94% pathogenic"

Without Bayesian thinking: "94% means it's definitely pathogenic!"

With Bayesian understanding: "94% is the posterior probability based on training data. But if this variant is very different from training examples—like a novel population or gene—the real uncertainty might be much higher. I should validate experimentally for important decisions."

Key questions to ask: – Was the model trained on data similar to mine? – How certain is the model really? (Does it give uncertainty estimates?) – Is this variant similar to what the model has seen before?

Understand Why More Data Helps

Remember the globe tossing? – **9 tosses:** Estimate p = 0.67, but wide uncertainty (0.45 – 0.85) – **100 tosses:** Estimate p = 0.71, narrow uncertainty (0.62 – 0.79)

In genomics:

```
Variant predictor trained on 1,000 variants:
→ Medium confidence, may miss rare patterns

Same predictor trained on 1,000,000 variants:
→ High confidence, better generalization

Why? More data = sharper posterior = more confident predictions!
```

This explains why: – Large datasets lead to better models – Models improve over time as more data accumulates – Rare variant prediction is harder (less training data)

Recognize When Models Fail

The out-of-distribution problem:

```
Training data: 95% European ancestry variants
Your sample: African-specific variant

Bayesian perspective:
"My model learned P(pathogenic | features, European data)
 But you're asking about P(pathogenic | features, African data)
 These can be very different!
 My uncertainty should be HIGH, but standard models won't tell you this."
```

What this means for you: – Be cautious when your data differs from training data – Check if training data matches your population/condition – When possible, use models trained on diverse datasets

Design Better Experiments

Bayesian thinking guides experimental design:

```
Goal: Find causal variant for disease

Step 1: Start with highest prior
→ Try exome sequencing (coding variants in known genes)

Step 2: Update based on results
→ Nothing found? Prior for coding variant decreases
→ Expand to whole genome (regulatory regions)

Step 3: Keep updating
→ Still nothing? Consider structural variants, epigenetics

Each negative result updates your beliefs about where to look next!
```

Practical example:

You're validating 100 predicted pathogenic variants but can only test 10 in the lab.

How to choose?

```
Rank by: (Model confidence) × (Biological plausibility) × (Clinical importance)

This combines:
- Posterior from model (computational evidence)
- Prior from biology (what we know about the gene)
- Value of information (which answer helps most)
```

This Bayesian approach maximizes what you learn from limited experiments.

The Big Picture:

Deep learning isn't a black box when you understand its Bayesian foundation: – **Loss function** = maximize likelihood of data – **Regularization** = encode prior beliefs – **Training** = find maximum posterior – **Predictions** = best guess given what the model learned

This understanding helps you: – Know when to trust predictions – Recognize model limitations – Design better experiments – Interpret results critically

---

Math Box: Bayes' Theorem and Components

You can skip this section and still understand the concepts! This is for those curious about the formal mathematics.

The Complete Formula

```
P(H|D) = [P(D|H) × P(H)] / P(D)

Where:
P(H|D) = Posterior - what we want to know
P(D|H) = Likelihood - how well H explains D
P(H) = Prior - what we believed before D
P(D) = Evidence - total probability of seeing D
```

Calculating Evidence (Normalization)

```
P(D) = Σ P(D|H ) × P(H )  for all hypotheses i
```

This ensures probabilities sum to 1.0!

Globe example:

```
P(6W,3L) = P(6W,3L | p=0) × P(p=0)
         + P(6W,3L | p=0.25) × P(p=0.25)
         + P(6W,3L | p=0.50) × P(p=0.50)
         + P(6W,3L | p=0.75) × P(p=0.75)
         + P(6W,3L | p=1) × P(p=1)

         = 0×0.2 + 27×0.2 + 1×0.2 + 729×0.2 + 0×0.2
         = 0 + 5.4 + 0.2 + 145.8 + 0
         = 151.4

Then:
P(p=0.75 | 6W,3L) = (729×0.2) / 151.4 = 145.8 / 151.4 = 96.3%
```

Binomial Likelihood

For globe-tossing (or any binary outcome repeated n times):

```
L(p | k successes in n trials) = C(n,k) × p^k × (1-p)^(n-k)

Where:
C(n,k) = "n choose k" = n! / (k! × (n-k)!)
       = Number of ways to arrange k successes in n trials
```

```
Example: 6 waters in 9 tosses
C(9,6) = 9! / (6! × 3!) = 362880 / (720 × 6) = 84

For p = 0.75:
L(p=0.75) = 84 × 0.75^6 × 0.25^3
          = 84 × 0.178 × 0.0156
          = 0.234 (23.4% chance of seeing this data if p=0.75)
```

Updating with Each Observation

Bayesian inference can be done sequentially:

```
Posterior after observation 1 becomes prior for observation 2:

After W:    P(p | W)
After W,L:  P(p | W,L)   P(L | p) × P(p | W)
                                ↑
                     This was posterior after W,
                     now becomes prior for L!
```

This is how your brain updates beliefs with each piece of evidence!

Summary

Key Takeaways:

1. **You already think like a Bayesian** – Your brain naturally combines prior beliefs with new evidence in daily situations: troubleshooting PCR, interpreting "easy exam" claims, evaluating gene expression changes, and determining if called variants are real
2. **Biology operates on probability distributions** – Gene expression, protein structures, evolution, and immune responses are all inherently stochastic, not deterministic
3. **The globe-tossing example illustrates core Bayesian concepts:**
   - **Prior:** Beliefs before observing data
   - **Likelihood:** How well each hypothesis explains observed data (the "garden of forking paths")
   - **Posterior:** Updated beliefs after observing data
   - **More data → sharper confidence**
4. **Deep learning is practical Bayesian inference:**
   - Loss function □ Negative log-likelihood
   - Regularization □ Prior
   - Training □ Finding maximum posterior (MAP)
   - Trade-off: Full distribution for speed and scalability
5. **Understanding this connection helps you:**
   - Interpret model confidence correctly
   - Recognize when predictions are reliable vs. uncertain
   - Design better training data
   - Combine evidence appropriately
   - Critically evaluate AI tools
6. **Foundation models leverage Bayesian principles:**

- Pre-training creates an informed prior
- Fine-tuning updates with task-specific data
- Enables learning from fewer examples

The Big Picture:

```
Your Biological Intuition (already Bayesian)
              ↓
Formalized in Bayes' Theorem
              ↓
Approximated by Deep Learning (for scale)
              ↓
Applied to Genomics (AlphaFold, DeepVariant, etc.)
```

Deep learning isn't magic—it's a practical, scalable implementation of reasoning you already do naturally!

"All models are wrong, but some are useful." – George Box

In biology, we work with probability distributions, not certainties. Bayesian inference and deep learning help us navigate this uncertainty at the scale that modern genomics demands.

Key Terms

- **Bayesian inference:** Updating beliefs about hypotheses using observed data via Bayes' theorem
- **Prior P(H):** Initial belief about a hypothesis before seeing data
- **Likelihood P(D|H):** Probability of observing data given a hypothesis is true
- **Posterior P(H|D):** Updated belief after observing data
- **Maximum A Posteriori (MAP):** Finding the single most probable hypothesis (what deep learning does)
- **Probability distribution:** Description of all possible values and their probabilities
- **Stochastic:** Involving randomness or probability (not deterministic)
- **Garden of forking paths:** Intuitive visualization of how data arise under different hypotheses
- **Evidence P(D):** Total probability of observing data across all hypotheses (normalization constant)
- **Foundation model:** Model pre-trained on massive data, creating an informed prior for downstream tasks

---

Test Your Understanding: Can You Answer These?

1. You're analyzing a rare variant (chr15:28,197,037 A>G) found in one ASD patient. The variant caller reports "95% confidence." Your lab mate says "95% confident means it's definitely real—let's publish!" What would you tell them using Bayesian reasoning?

Answer:

The 95% confidence is misleading without considering the **prior probability** that a variant is real.

Bayesian reasoning:

```
Posterior (is it real?)  Likelihood (evidence from reads) × Prior (how rare are real variants?)

Prior considerations:
• Most genome positions match reference (~99.9%)
• Real rare variants are very uncommon (~0.1% per position)
• Sequencing errors happen (~1-2% in difficult regions)

Additional checks needed:
```

```
1. Is this in a repetitive region? (increases error prior)
2. What's the read depth? (10× vs 100× matters!)
3. Is it in gnomAD? (external evidence updates posterior)
4. Does it segregate with disease in family? (genetic evidence)
```

The key insight: "95% confidence" from the caller is just the **likelihood** (how well this hypothesis explains the read data). But your true confidence (posterior) must also consider how **rare** real variants are (prior).

A 95% likelihood × low prior (rare variants) might only give ~60–70% true posterior confidence. You need validation!

Real-world lesson: Tools like DeepVariant give you one number, but you need to think Bayesianly about whether to trust it based on biological context.

2. You're doing RNA-seq on cancer vs. normal tissue (n=3 each). Gene X shows: Control = 50 TPM (±5), Cancer = 55 TPM (±8), p=0.04. Your advisor says "p<0.05, so it's differentially expressed—add it to the paper!" Why might this be wrong from a Bayesian perspective?

Answer:

The p-value only tells you the **likelihood** (probability of seeing this data if there's no difference), but ignores the **prior** (how likely is this gene to be truly differentially expressed?) and the **biological uncertainty**.

Why biology operates on probability distributions, not fixed values:

```
The measurements you see:
• Control: 50 TPM (±5) → Actually a distribution: [45, 48, 50, 52, 55]
• Cancer: 55 TPM (±8) → Even wider distribution: [47, 52, 55, 58, 63]

These distributions OVERLAP heavily!
• Some control samples: 52-55 TPM
• Some cancer samples: 47-52 TPM
• The "difference" might just be sampling biological noise
```

Bayesian considerations: 1. **Effect size:** Only 1.1× fold-change (55/50)—biologically meaningful? 2. **Sample size:** n=3 is tiny—huge uncertainty in the true distribution 3. **Prior:** Is Gene X in a pathway known to be altered in this cancer? If no prior evidence → should be more skeptical 4. **Biological variation:** ±5 vs ±8 shows high variability → distributions overlap substantially

Better approach: - Calculate **posterior probability** that true difference > 1.5× fold-change (biologically meaningful threshold) - Consider: "Given this data AND what we know about cancer biology, what's the probability this gene is truly differentially expressed?" - Validate with qRT-PCR or check in TCGA dataset (updating posterior with new evidence!)

Key principle: p-values ignore prior knowledge and biological context. Bayesian thinking asks: "Given everything we know, how confident should we really be?"

3. AlphaFold predicts a protein structure with "very high confidence" (pLDDT > 90). But when your experimentalist colleague tries to crystallize it, the structure is completely different. Using the Bayesian framework, explain what went wrong and how AlphaFold's training relates to prior beliefs.

Answer:

AlphaFold's prediction is a **MAP estimate** (maximum a posteriori—single best guess) based on what it learned during training. The error illustrates the **limitation of priors** learned from training data.

What AlphaFold actually does (Bayesian view):

```
Prior: Patterns learned from ~200,000 known protein structures in PDB
       (mostly from model organisms, crystallizable proteins, structured domains)

Likelihood: How well does each structure explain the amino acid sequence?
```

```
Posterior (pLDDT > 90): "Given training data, this structure is highly probable"
```

What went wrong—Out-of-Distribution Problem:

```
AlphaFold's "prior" (training data) was biased:
• Trained mostly on: Crystallizable proteins, stable folds, model organisms
• Your protein might be: Intrinsically disordered, membrane protein, human-specific

The "very high confidence" means:
"I'm 90% sure this is correct... IF your protein is like the ones I was trained on"

But your protein is OUT-OF-DISTRIBUTION:
• Different organism/condition than training data
• Post-translational modifications not in training
• Context-dependent folding (in vivo vs in vitro)
• Intrinsically disordered region (not well-represented in PDB)
```

Deep learning trade-off: – □ **Fast & scalable:** Can predict millions of structures – □ **Learns complex patterns:** Better than physics-based methods for typical proteins – × **Overconfident:** Doesn't know what it doesn't know (no uncertainty distribution) – × **Prior-dependent:** Only as good as training data

Lesson for genomics: When using AI tools (DeepVariant, AlphaFold, variant effect predictors): 1. Check if YOUR data matches the training distribution 2. "High confidence" ≠ "definitely correct"—it means "confident based on what I've seen before" 3. Validate experimentally, especially for novel contexts 4. Understand the prior (training data bias)

Key principle: Deep learning gives MAP estimates based on training data priors. Always check if your biological question matches what the model was trained on!

Coding Lab 2: Bayesian Inference with Globe-Tossing

Objective: Implement the globe-tossing example and explore Bayesian inference interactively.

What You'll Learn: – Calculate and visualize posterior distributions – See how beliefs update with each observation – Understand the effect of different priors – Explore the impact of sample size on confidence – Apply Bayesian reasoning to a genomics problem (variant calling)

Duration: 60–75 minutes

You'll implement: 1. Globe-tossing with different priors (uniform, informative) 2. Visualize posterior evolution with sequential data 3. Compare 9 tosses vs. 100 tosses 4. Apply to variant calling: Is this a real SNP or sequencing error?

Open Coding Lab 2 in Google Colab