Biology Meets Data

AI × Biotechnology — Student Learning Guide

WEEKS 1–2 · FOUNDATION · HIGH SCHOOL

■ LEARNING OUTCOMES

- Understand: What Is Bioinformatics?
- Understand: DNA, RNA & Proteins as Data
- Understand: Python for Life Sciences
- Understand: k-mer Feature Engineering

■ Duration	■ Level	■ Language	■ Track
Weeks 1–2	Foundation	Python 3.10+	High School

■ The human genome has ~3.2 billion base pairs. If printed, it would fill 3,000 books of 1,000 pages each.

■ Overview — Biology Meets Data

Bioinformatics is the art of applying computation to living systems. DNA, RNA, and proteins are just structured data — strings of characters that carry biological meaning. Once you see biology through the lens of data science, every machine learning tool becomes a microscope.

01. What Is Bioinformatics?

The intersection of biology, computer science, and statistics. Used for genome sequencing, drug discovery, precision medicine, and tracking viral mutations in real time (COVID-19 sequencing pipelines processed millions of genomes per week at peak).

Concept	Details
DNA → data string	A, T, G, C characters — 3.2 billion of them in your genome
Protein → feature vector	20 amino acid types become ML input features
Expression matrix	Rows = genes, Columns = samples, Values = activity levels
FASTA format	Plain text file: >sequence_id on one line, sequence on next

02. DNA, RNA & Proteins as Data

The central dogma describes information flow: DNA is transcribed to RNA, which is translated into proteins. Each step is a data transformation with its own alphabet and statistical properties.

Concept	Details
Adenine (A)	Pairs with Thymine (T) in DNA, Uracil (U) in RNA
Guanine (G)	Pairs with Cytosine (C) — G≡C bonds are stronger than A=T
GC Content	(G+C)/length × 100 — ranges from 25% (insects) to 75% (thermophiles)
Codon	3 nucleotides coding one amino acid — 64 codons, 20 amino acids

03. Python for Life Sciences

Python dominates bioinformatics for three reasons: readable syntax, a rich ecosystem (Biopython, NumPy, Pandas, PyTorch), and a vast open-source community constantly releasing new tools and datasets.

Concept	Details
Biopython	Parse FASTA/FASTQ, run BLAST, work with sequence objects

NumPy	High-performance arrays for large sequence matrices
Pandas	Labeled tables — the workhorse of exploratory data analysis
Matplotlib/Seaborn	Visualization: heatmaps, GC plots, PCA charts

04. k-mer Feature Engineering

k-mers are substrings of length k. Every possible 3-mer from ATGCGA: ATG, TGC, GCG, CGA. The frequency of each k-mer across a sequence creates a fixed-size feature vector — the foundation of genome ML models.

Concept	Details
k=3 (trinucleotides)	4^3 = 64 features — fast, effective for species classification
k=4 (tetranucleotides)	4■ = 256 features — more specific, captures codon context
fillna(0)	k-mers not present in a sequence get frequency = 0
Normalization	Divide counts by total k-mers to get frequencies in [0,1]

■ Complete Week 1 Analysis Script

```
from Bio import SeqIO
import pandas as pd
from collections import Counter
def analyze(record):
    seq = str(record.seq).upper()
    L = len(seq)
    # k-mer frequencies (k=3)
    kmers = Counter(seq[i:i+3] for i in range(L-2)
                    if 'N' not in seq[i:i+3])
    total = sum(kmers.values())
    feat = {k: v/total for k, v in kmers.items()}
    feat.update({
        "id": record.id,
        "length": L,
        "gc": (seq.count("G")+seq.count("C"))/L,
        "at": (seq.count("A")+seq.count("T"))/L,
    })
    return feat
records = list(SeqIO.parse("genome.fasta", "fasta"))
df = pd.DataFrame([analyze(r) for r in records]).fillna(0)
df.to_csv("features.csv", index=False)
print(df[["id","length","gc","at"]].head())
```

■ Quiz Preview — Check Your Understanding

These are sample questions from the Moodle graded quiz for this week. Complete the full 10-question quiz in your course LMS after reviewing the material.

Q 1 **What does GC content measure?**

✓ *Percentage of G and C bases in a DNA sequence*

Q 2 **What shape is a one-hot encoded 50-bp sequence?**

✓ *(50, 4) — one row per base, 4 columns for A/T/G/C*

Q 3 **For k=3, how many unique k-mers are possible?**

✓ *64 — because $4^3 = 64$ combinations of A,T,G,C*

Q 4	**Name two Python libraries used in bioinformatics.**
✓	*Biopython, NumPy, Pandas, scikit-learn, PyTorch (any 2)*

■ Resources & Next Steps

Type	Resource
■ Video	Course LMS → This week's video lesson (on-demand)
■ Dataset	Course LMS → datasets/ folder for this week
■ Project	Course LMS → project assignment notebook template
■ Quiz	Course LMS → Graded quiz (10 questions, timed)
■ Docs	See README.md in ai-biotech-portfolio GitHub repo