ChatGPT

Structured Learning Path: ML/DL in Biomedicine

Designing an effective curriculum requires decomposing the domain into prerequisite-linked modules (a “skill tree”) so that each concept builds naturally on earlier ones [1]. In the biomedical ML/DL context, foundational topics (math, programming, biology) underpin core ML techniques, which in turn enable specialized methods (see **Table 1**). This dependency-graph approach helps learners progress smoothly: students in a recent study noted that explicitly linking skills made the course “fit together well” and helped them find targeted practice when needed [1] [2]. We adopt this strategy to plan phases of study, from fundamentals through advanced research topics.

Module/Topic	Prerequisites
Programming & Data Tools	(None) – Python proficiency, libraries (numpy, pandas, scikit-learn, PyTorch/TensorFlow), version control, computing environment (Linux/GPU)
Math & Statistics Foundations	(None) – Linear algebra (vectors, matrices, eigen), calculus (derivatives, gradients), probability and statistics (distributions, hypothesis testing)
Biology & Genomics Basics	(None) – Molecular biology (DNA/RNA/proteins), genetics (genomes, variation), biomedical context (disease mechanisms)
Bioinformatics Data Preprocessing	Programming, Biology – NGS data formats (FASTA/FASTQ/VCF), quality control (FastQC), pipelines (alignment, normalization), tools (Biopython)
Machine Learning Fundamentals	Programming, Math – Supervised methods (regression, classification, SVM, trees), unsupervised (clustering, PCA/TSNE), model evaluation (cross-validation, metrics)
Basic Neural Networks	ML Fundamentals, Math – Perceptron and multi-layer perceptron, backpropagation, overfitting/regularization
Deep Learning Frameworks	Programming, Basic NN – TensorFlow/PyTorch, GPU usage, model training loops
Computer Vision & CNNs in Biomedicine	Deep Learning Frameworks – Convolutional nets, transfer learning on medical images (e.g. radiology, histopathology)
Sequence Models & Transformers	Deep Learning Frameworks, Genomics – RNN/LSTM and Transformer (attention) architectures for DNA/protein sequences, embeddings
Generative Models (VAEs/GANs) for Biology	Deep Learning Frameworks – Variational autoencoders, GANs for molecule and sequence generation (drug design, synthetic genomics)
Interpretable ML in Biomedicine	ML Fundamentals, Deep Learning – Feature importance (SHAP/LIME), saliency maps, concept attribution for biological models

Module/Topic	Prerequisites
Graph Neural Networks (GNNs) for Bio-graphs	ML Fundamentals, Deep Learning – Graph representations of molecular structures and knowledge graphs; GNN architectures for link prediction
Causal Inference in Healthcare	Statistics, ML Fundamentals – Causal graphs, counterfactuals, treatment effect estimation (Mendelian randomization, do-calculus)

Table 1. *Core modules and prerequisites in the dependency-graph learning system.* Each advanced module depends on mastery of its prerequisites in earlier modules. This mirrors competency-based curricula where clear prerequisite links "made the course more structured" and aided targeted practice [3].

Phase-Based Progression Plan

We propose a multi-phase learning plan, allocating roughly 4–8 weeks per phase (at ~16 h/week) and culminating in research-ready skills. Each phase builds on the previous one (per the graph in Table 1).

Phase 1 – Foundations (Months 1–2)

Goal: Acquire essential background in math, programming, and biology. These are prerequisites for all subsequent ML/DL topics.

- **Mathematical Foundations.** Build linear algebra (vectors, matrices, eigenvalues, SVD), calculus (gradients, optimization basics), and probability/statistics (distributions, hypothesis testing) skills.
- *Learning Objectives:* Compute matrix operations; understand gradients and loss minimization; grasp probability theory (Bayes' rule, common distributions).
- *Exercises:* Solve linear algebra problems (e.g. PCA from scratch); derive and implement gradients for simple cost functions; work through probability puzzles.
- *Resources:* MIT OpenCourseWare Linear Algebra and Multivariable Calculus (e.g. Strang's videos), Khan Academy (linear algebra, calc, stats modules); *The Elements of Statistical Learning* (Hastie et al.) chapters on linear models.

- *Assessment:* Correctly implement a simple PCA or regression with numpy and predict on synthetic data (e.g. reduce dimensions of a gene expression matrix).

- **Python & Data Tools.** Ensure fluency in Python for data analysis: NumPy, pandas, scikit-learn, Matplotlib/Seaborn for plotting, and a deep learning framework (TensorFlow or PyTorch). Learn best practices (Jupyter notebooks, Git, virtual environments, possibly Docker).

- *Objectives:* Manipulate arrays/dataframes; perform I/O (CSV, HDF5); build scikit-learn pipelines; use Git.
- *Exercises:* Reproduce a small Kaggle-style ML pipeline: load a dataset (e.g. UCI), preprocess features, train/test a model, and visualize results.
- *Resources: Python Data Science Handbook* (Vanderplas) or *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow* (Géron) [4]; scikit-learn tutorials; PyTorch tutorials or TensorFlow docs.

- *Assessment:* Build a reproducible ML pipeline (e.g. on a biomedical dataset from UCI or Kaggle) that loads data, trains a model, and saves results. Confirm it runs end-to-end on new data.

- **Biology & Genomics Basics.** Learn key biological concepts: DNA/RNA structure, genes and proteins, genetic variation (SNPs, mutations), basic cell biology and disease pathology. Understand types of data (sequence data, gene expression, proteomics, medical images, EHRs).

- *Objectives:* Explain how genomes encode information; describe central dogma (DNA→RNA→protein); know common genomic data formats and sources (e.g. FASTA, FASTQ, PDB, clinical records).
- *Exercises:* Obtain a small FASTA/FASTQ sample, parse it (e.g. using Biopython), and compute simple statistics (GC content, sequence length distribution).
- *Resources:* A short course/textbook on genomics or bioinformatics (e.g. Bioinformatics Data Skills by V. Buffalo for practical handling of genomic data); NCBI's Introduction to Genomics.

- *Assessment:* Given a raw sequencing file, produce a brief summary report of the data (e.g. number of reads, base quality summary). Confirm ability to annotate a gene name using an online database.

- **Bioinformatics Data Preprocessing.** Apply programming skills to real biological data. Learn NGS data processing (alignment, variant calling), and general data cleaning (handling missing data, normalization, one-hot encoding).

- *Objectives:* Use tools like Biopython, pybedtools; perform sequence alignment (e.g. with BWA or Bowtie) and process output (SAM/BAM). Understand normalization methods (RPKM/TPM for RNA-seq, batch correction).
- *Exercises:* Run a small RNA-seq or DNA-seq pipeline (e.g. align a FASTQ to a reference, call variants with bcftools, filter results). Normalize a gene expression matrix and check distribution.
- *Resources:* Official tutorials for FastQC and Samtools/Bcftools, Biopython documentation.
- *Assessment:* Produce a cleaned, analysis-ready dataset (e.g. variant VCF or normalized expression table). Check reproducibility by rerunning a pipeline script.

During Phase 1, the learner establishes a solid base: coding and math skills plus domain knowledge. As Greener *et al.* note, biological data are often large and complex, so these foundations are critical to avoid "pitfalls as well as opportunities" [5] .

Phase 2 – Core Machine Learning Techniques (Months 3–4)

Goal: Master general ML methods and practice on biological datasets. This phase follows the dependency graph so that concepts like supervised learning, unsupervised learning, and evaluation naturally build on the programming/math base.

- **Supervised Learning (Regression & Classification).** Cover linear/logistic regression, decision trees, SVMs, k-NN, ensemble methods (random forests, boosting).
- *Objectives:* Formulate prediction problems; implement and train basic classifiers/regressors; understand loss functions (MSE, cross-entropy); learn regularization (L1/L2, dropout); tune hyperparameters.
- *Exercises:* On a biomedical dataset (e.g. predicting patient outcome from gene expression or clinical features), implement and compare at least three different classifiers. Use cross-validation and grid search to optimize performance.
- *Resources:* Coursera ML course (Andrew Ng) or textbook sections; Scikit-learn user guide; survey of ML in biology (e.g. Greener *et al.* provide application examples [6]).

- *Assessment:* Achieve a specified accuracy or ROC AUC on a held-out biomedical classification task (e.g. cancer vs. normal tissue). Document the model selection process and metrics.

- **Unsupervised Learning & Dimensionality Reduction.** Learn clustering (k-means, hierarchical, DBSCAN) and techniques like PCA, t-SNE, UMAP.

- *Objectives:* Identify patterns without labels; reduce data to low dimensions for visualization/feature extraction.
- *Exercises:* Apply PCA/t-SNE to a gene expression matrix to find clusters of samples or genes. Cluster cells or patient profiles and interpret cluster characteristics.
- *Resources:* Scikit-learn tutorials; example (USC course) usage of PCA/t-SNE on scRNA-seq [7]; UMAP documentation.

- *Assessment:* Produce an informative plot (e.g. 2D PCA of single-cell data) where known classes (e.g. cell types) are distinguishable. Provide quantitative metrics (e.g. silhouette score) for clusters.

- **Model Evaluation & Pipelines.** Emphasize cross-validation, train/test splits, and avoiding data leakage. Learn to use `Pipeline` constructs and hyperparameter search.

- *Objectives:* Properly partition data (stratified splits), evaluate with appropriate metrics (e.g. precision/recall for imbalanced data), and integrate preprocessing/feature engineering steps into a single pipeline.
- *Exercises:* Create a full ML pipeline that imputes missing values, scales features, and trains a model. Use cross-validation to evaluate and compare models.
- *Resources:* Scikit-learn documentation (Pipelines, model selection); blogs on best practices.

- *Assessment:* Validate model generalization by cross-validation, reporting metrics (accuracy, F1, etc.) and confidence intervals. Demonstrate reproducing results after pipeline update.

- **Introduction to Neural Networks.** Move from classical ML to basic neural networks (multilayer perceptron).

- *Objectives:* Understand neuron units, activation functions, forward/backward propagation, and the concept of deep architectures.
- *Exercises:* Code a simple feedforward neural network from scratch (in NumPy) to solve a toy problem (e.g. XOR or a simple classification).
- *Resources:* Excerpts from Goodfellow *et al.* (Deep Learning textbook); online tutorials (3Blue1Brown's "Neural Networks" video series).
- *Assessment:* A simple network that converges to low training loss on the toy problem. Check gradients manually for a small example to ensure backprop works.

At the end of Phase 2, the learner should competently apply basic ML models to biomedical data. The course syllabus at USC, for example, emphasizes these very topics (regression, classification, clustering, PCA, ensemble methods, and introductory deep learning) [4], reflecting standard core ML curriculum. The emphasis on pipelines and cross-validation also aligns with best practices advised in the literature [5].

Phase 3 – Deep Learning & Domain Applications (Months 5–7)

Goal: Build deep learning expertise and apply it to rich biomedical domains such as images, sequences, and graphs.

- **Deep Learning Frameworks (TensorFlow/PyTorch).** Gain practical skills with a DL library.
- *Objectives:* Use GPU acceleration, define and train neural networks, handle data loaders, save/load models.
- *Exercises:* Re-implement the Phase 2 NN from scratch in a framework (e.g. as a PyTorch `nn.Module`), train on a real dataset (e.g. classify MNIST digits as a warm-up, then a simple biomedical image task).
- *Resources:* Official tutorials (TensorFlow/Keras or PyTorch); fast.ai practical course modules.

- *Assessment:* Successfully train and evaluate a model in the framework, and tune training (learning rate, batch size). Confirm reproducibility by fixing a random seed.

- **Convolutional Neural Networks for Biomedical Imaging.** Apply CNNs to medical images (radiographs, histology, X-rays, MRI/CT scans).

- *Objectives:* Understand convolution, pooling, and architectures (VGG, ResNet, UNet); use transfer learning with pretrained networks; address medical-specific challenges (3D data, class imbalance).
- *Exercises:* Use a public medical imaging dataset (e.g. chest X-ray pneumonia, retinopathy images, or the Brain Tumor Segmentation dataset) to train a classifier or segmentation model. Experiment with data augmentation and pretrained backbones.
- *Resources:* Stanford CS231n (convolutional nets), Kaggle tutorials (e.g. Diabetic Retinopathy Challenge); *nnU-Net* paper for segmentation benchmarks; CheXpert or NIH ChestX-ray datasets.

- *Assessment:* Achieve a reasonable performance (e.g. >85% accuracy or AUROC >0.85) on a held-out test, and visualize some model predictions. Use Grad-CAM or saliency to highlight regions influencing decisions.

- **Sequence Models (RNNs/Transformers) for Genomics.** Model biological sequences (DNA, RNA, proteins) with deep architectures.

- *Objectives:* Learn RNNs/LSTMs/GRUs and the Transformer (attention) architecture. Encode sequences (one-hot, embeddings); handle long sequence contexts.
- *Exercises:* Train an LSTM to predict a property of DNA sequences (e.g. promoter vs. non-promoter) or a Transformer (attention-based) model on genomic data. Use position encoding for sequences.
- *Resources:* Primer on deep learning in genomics [8] ; reviews of transformers in genomics (e.g. *Biology (Basel)* 2023) introducing attention for sequence analysis.

- *Assessment:* Demonstrate that the model learns sequence motifs (inspect attention weights or feature importances). Compare with a motif-search baseline and report metrics.

- **Autoencoders and Variational Autoencoders (VAEs).** Perform unsupervised representation learning.

- *Objectives:* Use autoencoders to compress data (images, gene expression); learn latent embeddings; understand VAE concepts (probabilistic latent space).
- *Exercises:* Apply a VAE to gene expression data (e.g. single-cell RNA-seq) to learn a low-dimensional latent representation. Visualize how different cell types cluster in latent space.
- *Resources:* TensorFlow/Keras or PyTorch guides for autoencoders; example project on single-cell autoencoders (e.g. scVI framework paper).

- *Assessment:* Reconstruction accuracy and latent visualization quality. Check that similar samples map near each other in latent space.

- **Generative Modeling for Biology.** Explore Generative Adversarial Networks (GANs) and other DGMs in biomedical contexts.

- *Objectives:* Understand GAN architecture (generator/discriminator) and applications to molecules or biological structures. Study VAE and flow-based generative models for molecule generation and synthetic genomics.
- *Exercises:* Replicate a published experiment: e.g. use a VAE to generate novel chemical structures or DNA sequences with desired properties, or use a GAN to augment imaging data.
- *Resources:* **Review:** López *et al.*, *Annual Rev Biodesign* 2023 [9] (overview of deep generative models in genomics, showing that DGMs can "generate novel genomic instances" with real-like characteristics). GAN tutorials for molecular design (e.g. ORGAN or MolGAN).
- *Assessment:* Evaluate generated samples with domain-specific criteria (e.g. chemical validity, novelty, similarity to training). Present results of a small generative model and discuss its limitations.

During Phase 3, the learner transitions to deep learning practice. The syllabus for an ML-in-biology course suggests many of these: CNNs on images, RNNs on sequences, and an introduction to deep networks [4]. Notably, deep generative models are increasingly used in drug discovery and genomics: DGMs "learn the complex structure of genomic data" and can synthesize realistic biological data [9]. Mastery here opens doors to creative research in drug design and synthetic biology.

Phase 4 – Advanced Specialized Topics (Months 8–10)

Goal: Delve into cutting-edge methods and interpretability, preparing for research. Focus on complex models (GNNs, attention) and rigorous analysis (causal inference, model explainability).

- **Interpretable ML for Biomedicine.** Study techniques to understand "black-box" models.
- *Objectives:* Learn model-agnostic methods (LIME, SHAP), CNN attribution (saliency maps, Grad-CAM), and inherently interpretable models (decision trees, generalized additive models). Understand trade-offs between accuracy and interpretability.
- *Exercises:* Apply SHAP to a complex model (e.g. a gene expression classifier) to rank gene importance. Use saliency maps to highlight image regions used by a CNN in diagnosis.
- *Resources:* Christoph Molnar's *Interpretable Machine Learning* (2020); tutorials on SHAP/LIME; domain papers on explainable AI in healthcare. Note that Greener *et al.* emphasize this need, observing "biologists want to know why a model makes a prediction," and that "interpreting a neural network… is generally much harder" [10].

- *Assessment:* Generate visual or quantitative explanations for at least two models (one tree-based, one neural) and evaluate their consistency (e.g. by checking if important features make biological sense). Present an interpretation as if to a domain scientist.

- **Graph Neural Networks for Biomedical Knowledge Graphs.** Leverage graph-based ML on biological networks.

- *Objectives:* Represent molecules as graphs (atoms/nodes, bonds/edges), and represent knowledge (genes, diseases, drugs) as heterogeneous graphs. Learn GNN variants (GCN, GraphSAGE, GAT).
- *Exercises:* Implement a GNN to predict drug–target interactions using a known dataset (e.g. DrugBank). Construct a small biomedical knowledge graph (e.g. diseases and symptoms) and perform node classification or link prediction.
- *Resources:* **Overview:** Choubisa *et al.*, 2024 [11] – "GNNs have emerged as a powerful approach, enabling the representation of molecules and biological interactions as graphs. This capability allows GNNs to capture complex relationships... facilitating the prediction of properties, drug–target interactions, and potential side effects." This review highlights GNN success in drug discovery. Use PyTorch Geometric or DGL libraries.

- *Assessment:* Report predictive performance of GNN vs baseline (e.g. a random forest on computed features). Visualize learned embeddings (e.g. using t-SNE) to show related entities cluster.

- **Attention Mechanisms & Transformers in Biology.** Study self-attention in depth.

- *Objectives:* Extend sequence modeling by mastering transformers and attention. Apply them to genomics (e.g. DNA language models) or biomedical text (clinical notes).
- *Exercises:* Fine-tune a pretrained transformer for a biomedical task (e.g. BERT-style model for protein function classification, or DNABERT for sequence classification). Compare to non-attention models.
- *Resources:* Review articles on transformers in genomics (e.g. *Biology (Basel)* 2023 for genome data); Vaswani *et al.* original transformer paper (for foundational understanding). Attention scores compute pairwise importance (dot-product + softmax) between sequence positions [12] .

- *Assessment:* Examine and interpret attention maps (e.g. identify motifs that attract high attention). Measure model performance gain from attention (e.g. longer-range dependencies captured).

- **Causal Inference in Healthcare.** Introduce causal reasoning and its relevance to biomedical research.

- *Objectives:* Learn causal diagrams, confounding vs. causation, and basic methods (propensity scores, instrumental variables, and do-calculus). Understand concepts like Mendelian randomization.
- *Exercises:* Formulate a causal graph for a clinical question (e.g. effect of a drug on an outcome with confounders). Use a simple dataset (possibly simulated) to estimate a causal effect with both naive and causal methods.
- *Resources:* Judea Pearl's *Causality* (Selected chapters); Hernán & Robins' *Causal Inference: What If*; review on causal ML in medicine (e.g. Royal Society Open Science, 2022).

- *Assessment:* Demonstrate correcting for a confounder in a toy problem or real clinical dataset. Explain the assumptions needed for causal claims.

- **Advanced Topics and Ethics.** Optionally explore federated learning (for privacy), reinforcement learning in medical decision-making, or advanced Bayesian methods. Ensure also to cover AI ethics (bias, fairness, data privacy) in healthcare.

- *Objectives:* Awareness of data privacy (HIPAA, de-identification), biases in medical AI, and regulatory considerations.
- *Exercises:* Read a case study on an AI diagnostic tool (e.g. pneumonia detection) and critique its methodology and fairness.
- *Resources:* “Ethical and Social Challenges of AI in Medicine” (X symposium papers), NIH/WHO guidelines on AI ethics, relevant chapters in ML textbooks.
- *Assessment:* Short essay or presentation on ethics considerations for one’s project.

Phase 4 consolidates expertise in specialized areas. For example, Choubisa *et al.* emphasize how GNNs enable “graph representations of molecules” for drug discovery [11]. Attention-based models are transforming genomics, with studies showing that self-attention can “interpret non-coding regions at base resolution.” Interpreting complex models is equally vital: the literature notes that deep networks are “not easily interpretable” for humans, so explainability tools become “useful as a sanity check” [10].

Phase 5 – Integration and Research Projects (Months 11–12)

Goal: Apply what’s learned in open-ended projects and prepare for independent research.

- **Capstone Projects / Paper Replication.** Select a recent research paper at the ML–biomed intersection (e.g. graph-based drug repurposing, deep learning on electronic health records) and replicate its key results or implement an extension.
- *Objectives:* Experience the research workflow: literature search, data acquisition, implementation, and evaluation.
- *Exercises:* Possible projects include: implementing the “DREAMwalk” knowledge-graph model for drug–disease links (Bang *et al.*, 2023); reproducing a high-impact genomics DL paper; or competing in a biomedical Kaggle challenge.
- *Resources:* Conferences/journals (NeurIPS, ICML, Nature Machine Intelligence, Bioinformatics), code repositories (GitHub) for state-of-art models. Use platforms like Google Colab or university clusters.

- *Assessment:* A written report and code demonstrating the replication. Compare your results with the paper, discuss discrepancies, and articulate biological insight gained.

- **Continuing Learning & Collaboration.** Build habits for self-directed research: follow ML in biomed journals, participate in community forums (Biostars, Cross Validated, Kaggle), and contribute to open-source bio-ML projects.

- *Exercises:* Set up alerts on bioRxiv/medRxiv for relevant keywords; join a journal club or local meetup on AI in healthcare.
- *Self-assessment:* At least one accepted submission to a conference or preprint by end of plan, or a polished GitHub project.

This final phase transitions the learner from student to researcher. It ensures each module has a tangible outcome (e.g. a mini-project or presentation) so the learner can “self-assess” mastery.

Learning Objectives, Exercises, and Resources per Module

Each module above has specific goals. For clarity, here is a concise summary (see also Table 2 for an example of one module's plan):

- **Learning Objectives:** What the learner should *know* or *be able to do*.
- **Exercises:** Practical projects (coding, data analysis, replication of a study) that reinforce skills.
- **Resources:** Textbooks, courses, tutorials, and key papers.
- **Self-Assessment Criteria:** How the learner verifies their understanding (e.g. achieving certain metrics, explaining results, passing a quiz).

In practice, as illustrated by Olson *et al.* [13], having hands-on projects (like Kaggle or Kaggle-like datasets) and writing up results are critical to solidify knowledge. The instructor or learner should check off objectives by successful completion of exercises and readiness to explain results.

For example, in the *Generative Models* module, objectives might include "understand the architecture of VAEs/GANs" and "apply a VAE to genomic data." An exercise could be "train a VAE on single-cell RNA-seq data to generate synthetic profiles." A resource would be the annual review by López *et al.* [9] for conceptual understanding. Self-assessment could be satisfied by demonstrating that generated samples capture key data features (e.g. similar variance and cluster structure).

Below is an illustrative module plan template:

Module	Learning Objectives	Exercises	Resources	Assessment
Example: Graph Neural Networks for Bio-graphs	– Understand graph data (molecules, networks) – Implement GNN layers (GCN/GAT) – Apply to drug-target or PPI graphs	– Build a GNN for drug–target interaction prediction – Use PyTorch Geometric to load a bio-graph and perform node classification	– Choubisa et al. review [11] – PyTorch Geometric tutorials – Example dataset (e.g. DrugBank)	– Achieve a meaningful metric (AUROC > 0.8) on link prediction – Visualize learned embeddings of nodes

Table 2. *Example detailed plan for the "Graph Neural Networks" module.* Similar tables/plans can be created for each module listed above.

Overall, this structured approach, inspired by the Math Academy dependency graph, ensures the learner sees how each new topic connects to prior ones. It balances theory (textbooks, papers) with practice (coding projects, data analysis) and includes meta-skills (self-assessment, ethics). By following this plan diligently, a motivated learner with the given profile will build strong ML/DL expertise tailored to biomedical research.

Sources: Our plan incorporates best practices from ML education and bioinformatics. For example, Greener *et al.* highlight the rising importance of ML in biology and the need to match methods to data types [5] [6]. Courses like USC's QBIO460 outline the very topics above [4]. Recent reviews (Zou *et al.*, 2019; López *et al.*, 2023 [9]; Choubisa *et al.*, 2024 [11]) show how deep learning and GNNs are applied in genomics and drug discovery. These informed the module choices and recommended resources.

---

[1] [2] [3] Structuring Competency-Based Courses Through Skill Trees
https://arxiv.org/pdf/2504.16966

[4] [7] QBIO460_Syllabus.docx
https://web-app.usc.edu/soc/syllabus/20243/12974.pdf

[5] [6] [8] [10] [13] A guide to machine learning for biologists
https://hfenglab.org/NRev21.pdf

[9] An Overview of Deep Generative Models in Functional and Evolutionary Genomics | Annual Reviews
https://www.annualreviews.org/content/journals/10.1146/annurev-biodatasci-020722-115651

[11] An overview of graph neural networks for molecular biology: Challenges and solutions
https://ijsra.net/sites/default/files/IJSRA-2024-1985.pdf

[12] Transformer Architecture and Attention Mechanisms in Genome Data Analysis: A Comprehensive Review
https://www.mdpi.com/2079-7737/12/7/1033