Generative Biology

This manuscript (permalink) was automatically generated from In-Vivo-Group/generative-biology@ed59f48 on June 14, 2024.

Authors

- **Alexander J. Titus** ✉
  0000-0002-0145-9564 · alexandertitus
  In Vivo Group, Washington, DC, USA; International Computer Science Institute, Berkeley, CA, USA · Funded by Grant TBD

- **Matthew E. Walsh**
  0000-0003-1514-7761 · mwalsh52
  Center for Health Security, Johns Hopkins Bloomberg School of Public Health, Baltimore, MD, USA; Department of Environmental Health and Engineering, Johns Hopkins Bloomberg School of Public Health, Baltimore, MD, USA

✉ — Correspondence possible via GitHub Issues or email to Alexander J. Titus <publications@theinvivogroup.com>.

Abstract

The rapid pace of progress in generative artificial intelligence (AI) techniques like deep learning, reinforcement learning, and transformer neural networks is transforming the life sciences and biomedicine. This living review paper provides an updatable, comprehensive overview and analysis of the latest literature on generative biology – the application of cutting-edge generative AI methods to accelerate insights and innovation across the life sciences and healthcare. All authors are welcome to contribute to this review via pull requests.

The review synthesizes key developments in using generative models for de novo biomedical discovery, design, and decision support. It examines techniques and applications including deep learning on omics data for personalized medicine, generative chemistry for drug development, protein structure prediction for molecular engineering, image synthesis for pathology, language models for clinical decision support, robotic simulation for prosthetics, and generative networks for cell programming.

The review highlights representative studies and benchmarks in each area while contextualizing progress, limitations, emerging best practices, and directions for future work. It also discusses social and ethical challenges raised by generative biology applications, such as compounding bias, system opacity, and dual-use risks, alongside proposed solutions.

As a living review, this paper will be continually updated as the field rapidly advances to provide researchers and practitioners with an up-to-date reference on the state of the art in employing generative AI to accelerate biomedicine for the collective good.

Executive Summary

The goal with be a 2 page TL;DR of the review after v1 is complete. Need more content.

Introduction

This is the start of the Generative Biology living review!

Computers, Algorithms and the Internet

1950s and 1960s: Early computers and algorithms

Computers were used in the early 1950s for population genetics calculations [1]. Notably, the inception of computational modeling in biology dates to the origins of computer science itself. British mathematician and logician Alan Turing, often referred to as "the father of computing", used primitive computers to implement a model of biological morphogenesis (the emergence of pattern and shape in living organisms) in 1952 [2]. At about the same time, a computer called MANIAC was used for measuring speculative genetic codes; it was originally built for weaponry research at the Los Alamos National Laboratory in New Mexico [3].

Computers were used for the study of protein structure by the 1960s, and other increasingly diverse analyses. These developments marked the rise of the computational biology field, stemming from research focused on protein crystallography, in which scientists found computers indispensable for carrying out laborious Fourier analyses to determine the three-dimensional structure of proteins [4,5].

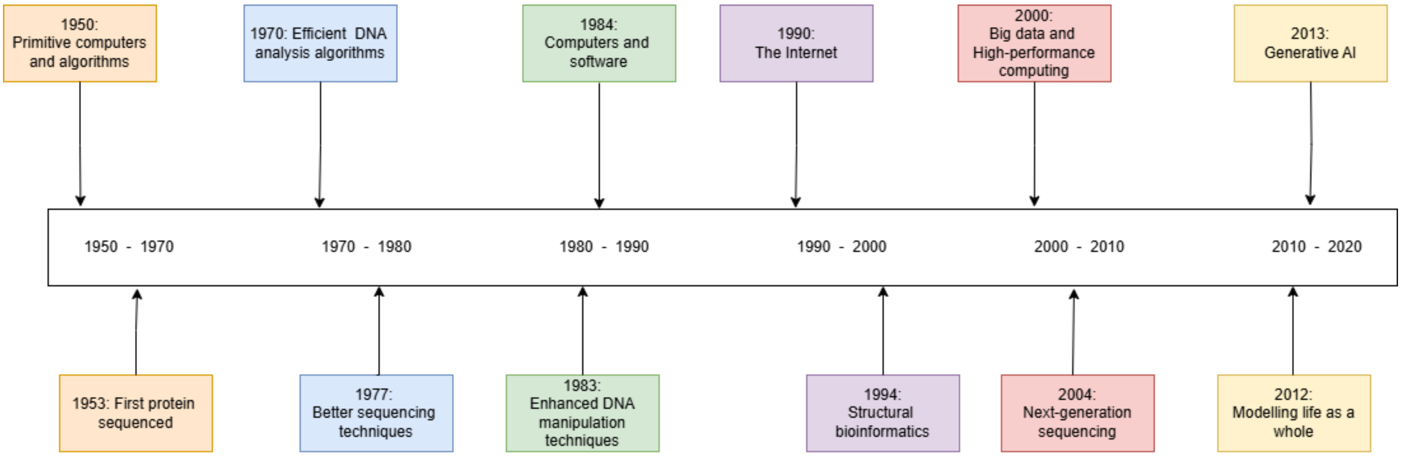
In addition to advances in determination of protein structures through crystallography, the first sequence of protein, insulin, was published [6,7]. More efficient protein sequencing methods, such as the Edman degradation technique [8], enabled sequencing 15 different proteins over a decade [9]. COMPROTEIN, one of the first bioinformatics softwares developed in the early 1960s, was designed to overcome the limitations of Edman sequencing [10]. In an effort to simplify the handling of protein sequence data for the COMPROTEIN software, a one-letter amino acid code was developed [11]. This one-letter code was first used in the Atlas of Protein Sequence and Structure [12], the first biological sequence database, laying the groundwork for paleogenetic studies.

Development of methods to compare protein sequences followed. The Needleman-Wunsch algorithm [13], the first dynamic programming algorithm developed for pairwise protein sequence alignments, was introduced in the 1970s. Multiple sequence alignment (MSA) algorithms followed in the early 1980s. Progressive sequence alignment was introduced by Feng and Doolittle in 1987 [14]. The MSA software CLUSTAL, a simplification of the Feng-Doolittle algorithm [15] was developed in 1988. It is still used and maintained to this day [16].

1970s: From protein to DNA analysis

The deciphering of all 64 triplet codons of the genetic code in 196817 fueled a desire to efficiently determine the sequence of DNA that existed into the 1970s. This desire led to the development of cost-efficient DNA sequencing methods, such as the Maxam-Gilbert and Sanger sequencing techniques in the mid-1970s [6,7,17]. With this new ability to generate DNA sequence data, a paradigm shift from protein analysis to DNA analysis occurred in the late 1970s. Concurrently, concerns over recombinant DNA research led to safety protocols established during the 1975 Asilomar conference [18].

New DNA sequencing techniques resulted in significantly more data to be analyzed, a task at which computation could help. The first software dedicated to analyzing Sanger sequencing reads was published in 1979 [19]. DNA sequences began to be utilized in phylogenetic inference with pioneering methods like maximum likelihood for inferring phylogenetic trees from DNA sequences [20]. Several bioinformatics tools and statistical methods were developed following this work. The adoption of Bayesian statistics in molecular phylogeny in the 1990s was inspired by this [21] and is still commonly used in biology today [22]. Yet, numerous computational limitations needed to be overcome during the latter half of the 1970s to expand the utilization of computing in the life sciences, especially in DNA analysis. The subsequent decade proved instrumental in addressing these challenges.



1950: Primitive computers and algorithms
1970: Efficient DNA analysis algorithms
1984: Computers and software
1990: The Internet
2000: Big data and High-performance computing
2013: Generative AI
1950 - 1970
1970 - 1980
1980 - 1990
1990 - 2000
2000 - 2010
2010 - 2020
1953: First protein sequenced
1977: Better sequencing techniques
1983: Enhanced DNA manipulation techniques
1994: Structural bioinformatics
2004: Next-generation sequencing
2012: Modelling life as a whole

Figure 1: The history of parallel advancements in computing and the life sciences: A timeline of major milestones.

1980s: Simultaneous advances in computing and biology

Parallel advancements in biology and computing propelled bioinformatics forward during the 1980s and 1990s. Molecular techniques like gene targeting and amplification, using enzymes like restriction endonucleases and DNA ligases, laid the groundwork for genetic engineering [18]. The polymerase chain reaction (PCR) transformed gene amplification, while innovations like Taq polymerase and thermal cyclers optimized the process [23].

Computing accessibility surged with microcomputers like the Commodore PET, Apple II, and Tandy TRS-80, along with bioinformatics software like the GCG software suite [24] and DNASTAR [25], another sequence manipulation suite capable of assembling and analyzing Sanger sequencing data. Other sequence manipulation suites were developed to run on CP/M, Apple II, and Macintosh computers [26] in the years 1984 and 1985. Free code copies of this software were offered on demand by some developers. This propelled an upcoming software-sharing movement in the programming world [27,28].

The free software movement, led by the GNU project, promoted open-source bioinformatics tools. Major sequence databases (EMBL, GenBank, DDBJ) standardized data formatting and enabled global sharing. Bioinformatics journals, like CABIOS, which is now known as Bioinformatics (Oxford, England) accentuated computational methods’ importance. Desktop workstations with Unix-like systems and scripting languages aided bioinformatics analyses, and scripting languages simplified tool development.

1990s: The genomics era and web-based bioinformatics

The genomics era began in the mid-1990s with the complete sequencing of the Haemophilus influenzae genome [29], initiating genome-scale analyses. This milestone was followed by the publication of the human genome at the beginning of the 21st century, which served as the definitive catalyst for the genomic era [30]. This transformative event spurred the design and development of several specialized Perl-based software to assemble whole-genome sequencing reads: PHRAP [31], Celera Assembler [32] among others.

Tim Berners-Lee's pioneering work at CERN in the early 1990s resulted in the World Wide Web, transforming global communication and ushering in an era of unprecedented access to information. With the advent of the internet, researchers gained a powerful platform to share and access vast amounts of biological data efficiently. This facilitated collaborative efforts in biology and genomics, leading to the establishment of foundational databases such as the EMBL Nucleotide Sequence Data Library [33] and the GenBank database became the responsibility of the NCBI [34] in 1992. Also, the famous NCBI website came online in 1994, featuring the efficient pairwise alignment tool BLAST [35]. After that, the world saw the birth of major databases we still rely on today: Genomes (1995), PubMed (1997), and Human Genome (1999) [36,37,38].

The proliferation of web-based resources transformed access to bioinformatics tools, democratizing their availability and usability for researchers worldwide. Through the development of web platforms, bioinformatics tools became more user-friendly and accessible. This shift enabled researchers to interact with sophisticated analytical tools without needing extensive computational expertise or access to specialized hardware. Consequently, the widespread adoption of web-based bioinformatics resources facilitated broader participation in genomic and molecular research, accelerating scientific discovery and collaboration on a global scale. Graphical web servers emerged as a convenient alternative to traditional UNIX-based systems, simplifying data analysis without the need for complex installations. The continued relevance of servers for scientific purposes is exemplified by the AlphaFold Server which uses the latest AlphaFold 3 model [39, J. et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3. Nature 1–3 (2024) doi:10.1038/s41586-024-07487-w], released in 2024, to provide highly accurate biomolecular structure predictions in a unified platform.

The internet facilitated the dissemination of scientific research through online publications, challenging traditional print-based methods. Early initiatives like BLEND [40] paved the way for internet-based scientific publishing by shedding insights into the potentials and obstacles associated with using the internet for scientific publications. This study paved the way for leveraging the Internet for both data set storage and dissemination, leading up to the establishment of preprint servers like arXiv (est. 1991) [41] and bioRxiv (est. 2013) [42] which changed the way scientific findings are shared and accessed. These platforms democratized access to scientific knowledge by enabling researchers to share their work rapidly and openly, facilitating interdisciplinary collaborations and the cross-pollination of ideas.

The experimental determination of the first three-dimensional structure of a protein, specifically, myoglobin, occurred in 1958 via X-ray diffraction [4]. However, earlier groundwork by Pauling and Corey with the publication of two articles in 1951 that reported the prediction of α-helices and β-sheets [43] laid the foundation for predicting protein structures. Similar to advances in other biological sciences, the utilization of computers has made it feasible to conduct calculations aimed at predicting the secondary and tertiary structure of proteins, with varying levels of confidence. This capability has been notably enhanced by the development of fold recognition algorithms, also known as threading algorithms [44,45]. However, proteins are dynamic entities, requiring advanced biophysical models to describe their interactions and movements accurately. Force fields have been formulated to describe the interactions among atoms, enabling the introduction of tools for modeling the molecular dynamics of proteins during the 1990s [46]. Used to study the behavior and interactions of atoms and molecules over time, molecular dynamics simulations calculate the positions and velocities of atoms based on physical principles. Despite the theoretical advancements and availability of tools, executing molecular dynamics simulations remained challenging in practice due to the substantial computational resources they demanded.

Graphical processing Units (GPUs) have made molecular dynamics more accessible [47], with applications extending to other bioinformatics fields requiring intensive computation. However, the internet's role in data dissemination, coupled with increasing computational power, has led to the proliferation of 'Big Data' in bioinformatics.

2000-2010: High-throughput sequencing and big data

Second-generation sequencing technologies democratized high-throughput bioinformatics. For example '454' pyrosequencing, a high-throughput DNA sequencing technique played a significant role in advancing genomics research by enabling rapid and cost-effective sequencing of DNA samples, particularly for applications such as whole-genome sequencing [48], but computational challenges arose with increased data volumes. Decreasing sequencing costs resulted in more data being generated, emphasizing data organization and accessibility. Specialized repositories and standardization efforts were needed to ensure data interoperability. High-performance computing adaptation became vital to address the increased amounts of data within bioinformatics projects. The surge in bioinformatics projects, accompanied by a vast influx of data, prompted adjustments from funding bodies to accommodate the demand for high-performance computing resources and collaborative initiatives.

While basic computer setups suffice for some projects, others demand complex infrastructures and substantial expertise. Government-sponsored entities like Compute Canada, New York State's High-Performance Computing Program, The European Technology Platform for High-Performance Computing, and National Center for High-Performance Computing served researchers' computational needs. Companies like Amazon, Microsoft, and Google, among many others, offer bioinformatics and life sciences services, emphasizing the field's importance.

Table 1. Organizations providing High-Performance Computing Resources for Bioinformatics and Life Sciences

Organization	Computing Resources
Compute Canada	Provides high-performance computing resources and support services to researchers and innovators across Canada. They offer supercomputers, cloud platforms, data storage, and training programs to advance scientific research and innovation in various fields.
New York State's High-Performance Computing Program	Provides researchers, businesses, and educational institutions with access to high-performance computing (HPC) resources and expertise to support their computational research and development efforts.
The European Technology Platform for High-Performance Computing	Fosters collaboration among industry, research, and academic stakeholders to advance high-performance computing (HPC) technology in Europe.
National Center for High-Performance Computing	Facility for high-performance computing (HPC) resources including large-scale computational science and engineering, cluster and grid computing, middleware development, visualization and virtual reality, data storage, networking, and HPC-related training.

Organization	Computing Resources
National Center for Supercomputing Applications	Offers high-performance computing resources such as the Blue Waters supercomputer, provides advanced data storage solutions, data analysis, and visualization tools, and supports interdisciplinary research in fields such as astrophysics, climate modeling, and genomics.
Oak Ridge Leadership Computing Facility	Provides supercomputing resources, such as the Summit supercomputer, for scientific research, offers support services including software development, data storage, and visualization, and facilitates research in various fields including climate science, biology, and materials science.
Swiss National Supercomputing Centre	Provides high-performance computing systems including the Piz Daint supercomputer, offers cloud computing services, data management, and user support, and facilitates scientific research in areas such as climate modeling, physics, and life sciences.
Barcelona Supercomputing Center	Provides access to MareNostrum, one of the most powerful supercomputers in Europe, offers resources for high-performance computing, data storage, and computational sciences, and supports research in fields including bioinformatics, computational biology, and engineering.
Japan's RIKEN Center for Computational Science	Houses the Fugaku supercomputer, one of the world's fastest supercomputers, provides resources for computational science, data processing, and artificial intelligence, and supports research in fields such as life sciences, materials science, and disaster prevention.
National Supercomputing Centre Singapore	Provides high-performance computing resources and support services, offers data storage, cloud computing, and software development services, and supports research in fields including bioinformatics, environmental modeling, and smart cities.

Community computing platforms democratized participation and expanded bioinformatics research's reach. Platforms like BOINC [49] enabled broad participation in bioinformatics. Experts can submit computing tasks to BOINC, while non-experts and science enthusiasts can volunteer their computer resources to process these tasks. Several life sciences projects are available through BOINC, including protein-ligand docking, malaria simulations, and protein folding [49].

2010-Today: The present and future

The integration of computers into biology has ushered in a new era of research possibilities, allowing for increasingly complex studies. While before, the focus was on individual genes or proteins, advancements today enable the analysis of entire genomes or proteomes [50]. This shift toward a holistic approach in biology is evident in disciplines like genomics, proteomics, and glycomics, which have limited interconnection between them.

The next leap at the intersection of computing and the life sciences lies in modeling entire living organisms and their environments simultaneously, integrating all molecular categories. This has already been achieved in a whole cell model of Mycoplasma genitalium, in which all its genes, products and their known metabolic interactions have been reconstructed [51]. Driven by advancements in measurement techniques, improved computational performance and artificial intelligence (AI) techniques, whole-cell modeling is increasingly becoming realistic and feasible. In contrast to traditional bottom-up approaches relying on molecular interaction networks, a predictive model has been developed for genome-wide phenotypes of budding yeast using deep learning [52]. The main applications of whole-cell modeling have been in producing useful substances and discovering drugs, such as antimicrobials [53,54,55,56] since whole-cell modeling was first directed toward unicellular organisms. Meanwhile, models of cultured human cells have also been developed, which have found applications in cell differentiation and medical research [57]. The possibility of modeling entire multicellular organisms may not be far off, considering the rapid pace of technological and computational advancements like artificial intelligence (AI) .

Artificial Intelligence (AI)

Artificial intelligence (AI) refers to a set of tools, techniques and paradigms that enable computers to mimic human behavior and either replicate the decision-making process typically performed by humans or exceed human performance in solving complex tasks independently or with minimal human intervention [58]. AI is concerned with a variety of central problems, including knowledge representation, reasoning, learning, planning, perception, and communication. It also refers to a variety of tools and methods, including case-based reasoning, rule-based systems, genetic algorithms, fuzzy models, and multi-agent systems [59]. Early AI research focused primarily on hard-coded statements in formal languages, which a computer can then automatically reason about based on logical inference rules. These computer systems known as expert systems, excelled in specific domains but lacked adaptability. Over time, AI has evolved to include a variety of approaches, each with its own strengths and weaknesses. For instance, expert systems are highly accurate within narrow fields but struggle with tasks outside their programmed knowledge. In contrast, machine learning algorithms can generalize from data and adapt to new situations, though they require large datasets and extensive training. Other AI techniques, such as deep learning, neural networks, and natural language processing also offer their own unique advantages and challenges.

Expert systems

Expert systems are a type of artificial intelligence (AI) that aims to replicate the decision-making capabilities of human experts in specific domains. They are made of a knowledge base containing domain-specific facts, rules, and heuristics, and an inference engine that applies logical reasoning to this knowledge to draw conclusions or make decisions [60]. Users are typically able to input queries and receive advice or recommendations through a simplified user interface. The primary user action, which involves pointing and clicking, is known as selecting [61].

An expert system for chemical analysis was developed in 1965 by AI researcher Edward Feigenbaum and geneticist Joshua Lederberg. This system was originally known as Heuristic DENDRAL and later as DENDRAL [62]. DENDRAL was developed to analyze molecular structures, particularly those containing elements like carbon, hydrogen, and nitrogen, based on spectrographic data. It proposed molecular structures for the compounds, with accuracy comparable to that of expert chemists.

Edward Shortliffe's work on MYCIN [63] began in 1972 at Stanford University. MYCIN, an expert system, was designed to assist physicians in diagnosing and selecting therapies for patients with bacterial infections, particularly patients with meningitis. It used a rule-based system that analyzed patient symptoms and medical history to suggest appropriate antibiotic treatments. MYCIN exhibited proficiency equivalent to infectious disease doctors.

However, despite their capabilities, the paradigm faces several limitations as humans generally struggle to explicitly articulate all their tacit knowledge that is required to perform complex tasks [64], leading to challenges such as difficulty in extrapolation, handling out-of-distribution data, managing uncertainty, and addressing biases. These limitations arise because expert systems heavily rely on predefined rules and knowledge encoded by humans. Consequently, the involvement of humans in specifying these parameters is essential but can also introduce limitations due to human cognitive constraints and biases. In contrast, machine learning algorithms overcome some of these limitations by learning from data, and making them more adaptable without relying heavily on explicit human guidance.

Machine learning and Deep learning

Machine learning (ML) is a subset of AI that focuses on the development of algorithms and statistical models that enable computers to perform tasks without being explicitly programmed to do so [65]. It involves the use of data and algorithms to imitate the way humans learn, gradually improving the system's performance on a specific task over time through iterative learning processes. Machine learning is effective for tasks such as classification, regression, and clustering, particularly when they involve high-dimensional data. These algorithms analyze data, identify patterns, and make predictions or decisions without being explicitly programmed for each task.

Based on the given problem and the available data, there are many potential model and training paradigms, three of the most prominent types of ML being: supervised learning [66], unsupervised learning [**url?** https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5765025,67], and reinforcement learning [68]. The goal of machine learning is to develop an output model that can make predictions or decisions based on input data. In supervised learning, the model is trained on a labeled dataset, where each training example is paired with an output label. A label is the desired output or result for a given piece of data. For example, in an image recognition task, labels could be the names of objects in the images (e.g., "cat," "dog," "car"). In a spam detection task, emails could be labeled as "spam" or "not spam.". The goal is to learn a mapping from inputs to outputs. Unsupervised learning involves training a model on data without labeled responses. The goal is to uncover patterns or structures within the data. In reinforcement learning, an agent learns to make decisions by interacting with an environment. The agent receives feedback in the form of rewards or penalties based on its actions and learns to maximize cumulative rewards over time.

Depending on the learning task, the field offers various classes of ML algorithms, each of them coming in multiple specifications and variants, including regression models, instance-based algorithms, decision trees, Bayesian methods, and artificial neural networks, among others.

Artificial neural networks (ANNs) span all three major types of machine learning. ANNs are inspired by biological systems and consist of interconnected processing units called neurons, with connections akin to synapses in the human brain. Signals are processed based on thresholds set by activation functions, and organized into layers for input, hidden, and output layers. Shallow machine learning encompasses simpler ANNs and other algorithms, often being more interpretable than deep neural networks. Deep neural networks, which have multiple hidden layers, perform complex calculations to automatically discover patterns in data. This ability is known as deep learning64. Deep learning excels with large, high-dimensional data like text, images, and videos, while shallow learning may outperform with low-dimensional data or limited training data. Time series, image, and text data present various application domains.

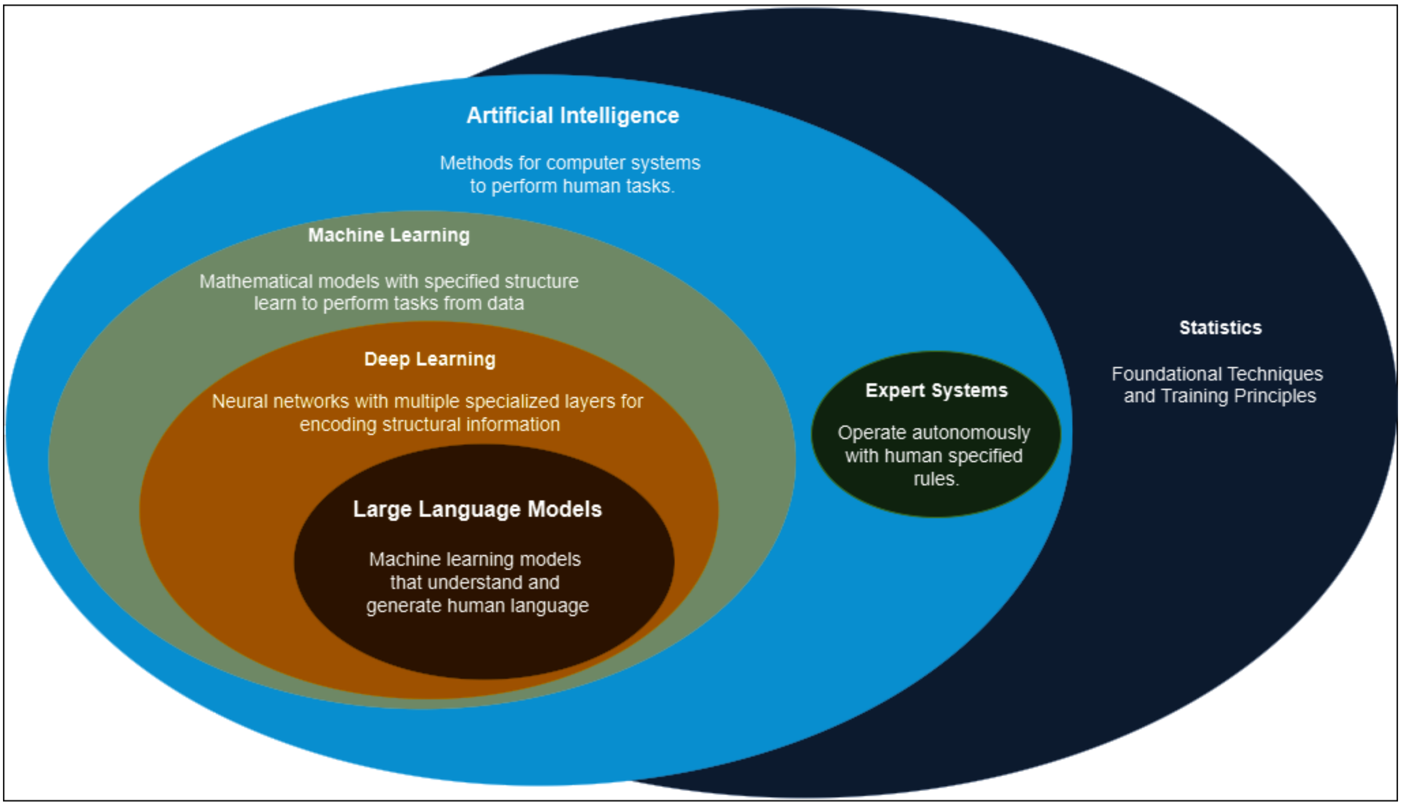

Artificial Intelligence
Methods for computer systems to perform human tasks.
Machine Learning
Mathematical models with specified structure learn to perform tasks from data
Deep Learning
Neural networks with multiple specialized layers for encoding structural information
Large Language Models
Machine learning models that understand and generate human language
Expert Systems
Operate autonomously with human specified rules.
Statistics
Foundational Techniques and Training Principles

Figure 2: Relationship between statistics, artificial intelligence, expert systems, machine learning, deep learning and large language models.

Automated model building in machine learning involves using input data for pattern identification relevant to the learning task. Shallow machine learning relies on predefined features such as pixel values in images or word frequencies in text. For example, in image classification, shallow learning might rely on handcrafted features like color histograms or edge detectors. In contrast, deep learning can operate directly on high-dimensional raw input data, such as the raw pixel values of an image or the sequence of words in text. It automatically learns features at multiple levels of abstraction, allowing it to capture patterns in the data without the need for manual feature engineering. For instance, in image classification with deep learning, the model learns to detect edges, shapes, and textures from raw pixel data, resulting in improved accuracy [69].

Deep learning architectures often combine both aspects into end-to-end systems or extract features for use in other learning subsystems. Various deep learning architectures have emerged, including convolutional neural networks (CNNs) [70], recurrent neural networks (RNNs) [71], distributed representations [72], autoencoders [73], generative adversarial neural networks (GANs) [74], among others. CNNs excel in computer vision and speech recognition tasks, learning hierarchical features essential for image recognition. RNNs specialize in sequential data structures like time-series data and natural language processing (NLP), addressing the challenges of vanishing gradients through advanced mechanisms like long short-term memory (LSTM) networks [75]. Distributed representations, such as word embeddings, play a crucial role in NLP tasks by projecting language entities into numerical representations, preserving semantic relationships between words. Autoencoders provide dense feature representations and are applied for unsupervised feature learning, dimensionality reduction, and anomaly detection. GANs, belonging to generative models, learn probability distributions over training data to generate new data samples, using a generator-discriminator framework in a non-cooperative game setting.

Generative AI and Transformers

Generative AI (GenAI) analyzes vast amounts of data, looking for patterns and relationships, then uses these insights to create fresh, new content that mimics the original data [76]. It does this by leveraging machine learning models, especially unsupervised and semi-supervised algorithms. There are three popular techniques for implementing Generative AI: Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and Transformers.

Variational Autoencoders (VAEs) [77] first introduced by Diederik P. Kingma et al. in 2013 are generative models in unsupervised machine learning that generate new data similar to the input data. They consist of an encoder that compresses the input data into a lower-dimensional latent space by producing parameters for a probability distribution (mean and variance). The decoder reconstructs the data from this latent representation. The loss function, which combines reconstruction loss and regularization loss (KL Divergence), ensures the output data is both accurate and diverse. VAEs are used in applications like image generation, data imputation, anomaly detection, offering a flexible framework for generating and understanding data despite some challenges in balancing the loss components and achieving high-quality outputs [78,79,80].

In 2014, GANs [74] were proposed by researchers at the University of Montreal. GANs use two models that work in tandem: One learns to generate a target output (like an image) and the other learns to discriminate true data from the generator's output. The generator tries to fool the discriminator, and in the process learns to make more realistic outputs. The image generator StyleGAN [81] is based on these types of models.

Diffusion models [82] were introduced a year later by researchers at Stanford University and the University of California at Berkeley. By iteratively refining their output, these models learn to generate new data samples that resemble samples in a training dataset and have been used to create realistic-looking images. A diffusion model is at the heart of the text-to-image generation system Stable Diffusion [83].

Recurrent neural networks (RNNs) and their variants like long short-term memory (LSTM) networks are commonly used for sequential data processing tasks. However, these models suffer from limitations such as vanishing gradients and inefficiency in parallelization. Transformers revolutionized the field with the ability to capture long-range dependencies in sequential data efficiently and was first reported in the seminal 2017 paper, “Attention is All You Need” [84]. The introduction of transformers, with their superior performance and scalability, initiated a departure from RNNs. Transformers were used to train the large language models (LLMs) that power ChatGPT [85].

The transformer architecture consists of an encoder and a decoder, each with multiple layers of self-attention and feedforward neural networks. The self-attention mechanism enables the model to assess the significance of a piece of data, such as a word in a sentence, based on that word's relations with other words in the sentence. To preserve the ordering of the words and the meaning of the sentence, the transformer incorporates positional bias to maintain the relative positions of words within a sentence.

The transformer encoder-decoder architecture performs well at tasks like language translation. In a language translation task, the model transforms a sentence by encoding inputs from one language and then decoding outputs in another. The encoder processes the input sentence and creates a fixed-size vector representation, which the decoder then uses to generate the output sentence. The encoder-decoder employs both self-attention and cross-attention mechanisms, where self-attention is applied to the decoder's inputs, and cross-attention focuses on the encoder's output.

A prominent example of the transformer encoder-decoder architecture is Google's T5 (Text-to-Text Transfer Transformer) [86], introduced in 2019. T5 can be fine-tuned for various NLP tasks, including language translation, question answering, and summarization.Real-world applications of the transformer encoder-decoder architecture include Google Translate, which utilizes the T5 model for translating text between languages, and Facebook's M2M-10080, a multilingual machine translation model capable of translating among 100 different languages.

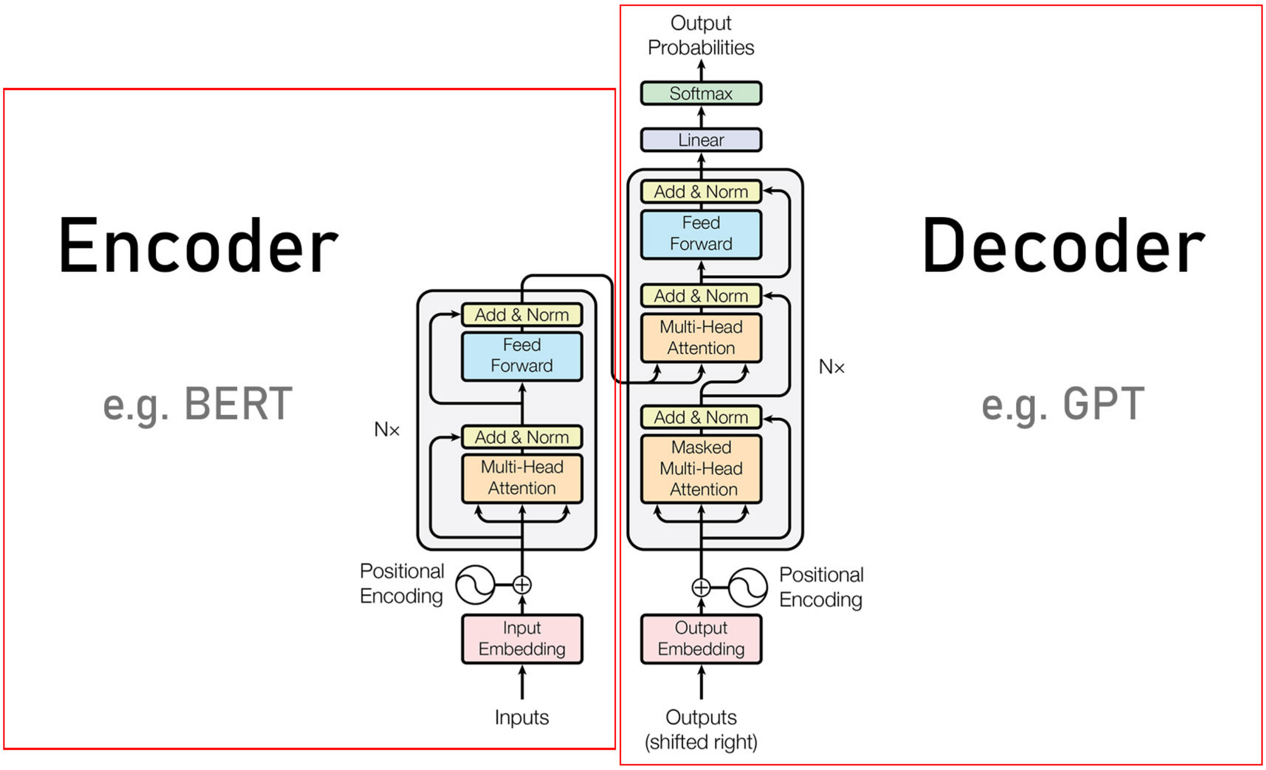


Figure 3: The encoder-decoder structure of the Transformer architecture. Adapted from "Attention Is All You Need" **Encoder-only models**: Ideal for tasks requiring a deep understanding of the input, such as sentence classification and named entity recognition. **Decoder-only models**: Suited for generative tasks like text generation. **Encoder-decoder models (or sequence-to-sequence models)**: Best for generative tasks that depend on an input, such as translation or summarization.

Transformer Encoder

The transformer encoder architecture is used for tasks such as text classification, where the goal is to categorize a piece of text into predefined categories. Text classification tasks include determining the sentiment of a piece of text, determining the topic and detecting if the text is spam. The encoder processes a sequence of tokens and produces a fixed-size vector representation of the entire sequence, which is then used for classification. The most notable transformer encoder model is BERT (Bidirectional Encoder Representations from Transformers) [87], introduced by Google in 2018. BERT is pre-trained on large text datasets and can be fine-tuned for a wide range of NLP tasks.

Unlike the encoder-decoder architecture, the transformer encoder focuses solely on the input sequence without generating an output sequence and instead the output is a classification task. It uses the self-attention mechanism to identify the most relevant parts of the input for the given task. Real-world applications of the transformer encoder architecture include sentiment analysis, where models classify reviews as positive or negative, and email spam detection, where models classify emails as spam or not.

Transformer Decoder

The transformer decoder architecture is tailored for tasks like language generation, where the model creates a sequence of words based on an input prompt or context. The decoder takes a fixed-size vector representation of the context and generates a sequence of words one at a time, with each word depending on the previously generated words. A well-known transformer decoder model is GPT-

3 (Generative Pre-trained Transformer 3) [88], introduced by OpenAI in 2020. GPT-3 is a large language model capable of generating human-like text across various styles and genres. ChatGPT, which is based on the GPT-3 model, was officially launched by OpenAI in November 2020. It was a significant milestone in the development of large language models (LLMs), characterized by its ability to generate human-like text across various styles and genres. Real-world applications of the transformer decoder architecture include text generation, where models generate stories or articles based on a given prompt, and chatbots, where models create natural and engaging responses to user inputs.

Large Language Models (LLMs)

Large language models are machine learning models that can comprehend and generate human language text. In the life sciences, LLMs such as GPT (Generative Pre-trained Transformer) and BERT, have revolutionized natural language processing, enabling researchers to extract insights from vast repositories of biomedical literature, accelerate drug discovery, and personalize patient care [89].

Large language models use transformer models and are trained using massive datasets — hence, large. This enables them to recognize, translate, predict, or generate text or other content. They are composed of multiple neural network layers – recurrent layers, feedforward layers, embedding layers, and attention layers work in tandem to process the input text and generate output content.

There are three main kinds of large language models:

- **Generic or raw language models** predict the next word based on the language in the training data. These language models perform information retrieval tasks.
- **Instruction-tuned language models** are trained to predict responses to the instructions given in the input. This allows them to perform sentiment analysis, or to generate text or code.
- **Dialog-tuned language models** are trained to have a dialog by predicting the next response. Think of chatbots or conversational AI.

Before functioning, LLMs undergo two crucial processes: training and fine-tuning. They are pre-trained on massive textual datasets from sources like Wikipedia and GitHub, comprising trillions of words to form a foundation model or a pre-trained model. This unsupervised learning stage allows the model to understand word meanings, relationships, and contextual distinctions, such as discerning whether “right” means “correct” or the opposite of “left.”. To perform specific tasks, pretrained models undergo fine-tuning, which tailors them to particular activities like translation. This process optimizes task-specific performance. A related method, prompt-tuning, trains the model using few-shot or zero-shot prompting. Few-shot prompting provides examples to teach the model how to respond, while zero-shot prompting directly instructs the model on the task without examples.

LLMs serve various purposes:

- **Information retrieval**: Used by search engines like Google and Bing to produce and communicate answers conversationally.
- **Sentiment analysis**: Used to evaluate the sentiment of textual data.
- **Text generation**: Powers generative AI, such as ChatGPT, to create text based on prompts.
- **Code generation**: Similar to text generation, LLMs can generate code by recognizing patterns.
- **Chatbots and conversational AI**: Facilitate customer service interactions by interpreting and responding to customer queries.

Advances in Generative AI for RNA

Introduction

- Background on key roles of RNA in biology
- Promise of generative models to advance RNA research
- Scope focused on latest advances in RNA prediction and design

Structure Prediction

- Transformer-based prediction of RNA folding
- Novel architectures for incorporating chemical constraints
- Improved accuracy on complex structures like ribosomes

Function Prediction

- Inferring RNA functions from sequence and structure
- Identifying motifs, domains, and atomic binding sites
- Applications in understanding long noncoding RNAs

Interaction Prediction

- Graph neural networks for RNA-protein interactions
- Structure-augmented modeling of splice sites
- Predicting RNA base editing targets

Design of RNA Therapeutics

- Generative models for optimizing siRNA, antisense design
- Reinforcement learning for chemical modification patterns
- Progress in computational RNA-targeted drug design

Outlook

- Key challenges in prediction of long RNA structures
- Design of RNA for self-assembly and scaffolding
- Ethical use of synthetic RNA technologies

Advances in Generative AI for DNA

Introduction

- Background on the role of DNA as a carrier of genetic information
- Promise of generative models to advance DNA research
- Scope focused on the latest advances in DNA prediction and design

Sequence Modeling

Transformer architectures for modeling DNA sequences

The HyenaDNA paper introduces a new genomic foundation model called HyenaDNA that can process DNA sequences at single nucleotide resolution with ultralong context lengths up to 1 million base pairs - a 500x increase over previous transformer models. HyenaDNA uses a parameter-efficient convolutional architecture that allows it to scale subquadratically with sequence length, enabling the use of full genome-scale context. On a range of regulatory genomics prediction tasks, HyenaDNA matches or exceeds the performance of previous state-of-the-art models while using 1500x fewer parameters and 3200x less pretraining data. HyenaDNA also demonstrates the ability to perform challenging species classification using the full mutational profile visible at 1 million base pairs of context. The authors explore new training techniques to enable ultralong sequence modeling as well as prompt-based tuning methods for rapidly adapting to new tasks without updating pre-trained weights [90].

Pretraining on large genomic datasets

Applications in variant calling and annotation

Regulation Prediction

- Graph neural networks for modeling 3D genome architecture
- Predicting enhancer-promoter interactions and expression
- Design of synthetic promoters and enhancers

Genome Editing

- Generative models for CRISPR guide design
- Contextual prediction of on-target editing efficacy
- Modeling of off-target effects during optimization

DNA Data Generation

- Variational autoencoders for realistic DNA sequences
- Generating paired genomic-transcriptomic data
- Applications in training genome interpretation models

Outlook

Challenges in predicting long-range chromatin interactions

The development of transformer architectures like HyenaDNA that can leverage genome-scale context creates new opportunities for understanding long-range chromatin interactions, gene regulation, and intercellular networks. However, significant challenges remain in improving model accuracy and uncertainty quantification. Hybrid physics- and data-driven approaches may help address these gaps. Safety considerations around synthetic genome design also warrant further research to ensure responsible innovation as these generative capabilities advance. Overall, the future looks bright for generative models that can capture both local mutations and global patterns critical to biology [90].

Responsible design of synthetic genomes

Ethical considerations for human genome editing

Generative AI for Autonomous Experimentation

Introduction

- Promise of generative models to accelerate scientific discovery
- Rise of autonomous labs and robotics for automated experimentation
- Overview of generative AI's role in self-driving research

Closed-Loop Systems

- Integrating computational hypothesis generation with robotic wet lab testing
- Reinforcement learning pipelines for autonomous optimization
- Case studies in materials science, drug discovery

Automated Experiment Design

- Using generative models to design novel compounds, genes
- Leveraging simulations to predict experimental outcomes
- Robotic execution of designed experiments

Adaptive Sampling

- Active learning to iteratively select most informative experiments
- Bayesian optimization powered by neural networks
- Applications in probing molecular design spaces

Real-Time Learning

- Deploying models on lab edge devices
- Online learning from experimental data streams
- Improving models and experiment plans on-the-fly

Outlook

- Key challenges around model accuracy and integration
- The future of data-driven, self-driving laboratories
- Risks of full automation and need for human oversight

Conclusion

- Generative AI as a powerful tool for autonomous experimentation
- Accelerating discovery alongside human researchers
- Responsible implementation will maximize benefit

Generative AI and the Biosecurity Landscape

Introduction

- Background on biosecurity threats from natural, accidental, and intentional pathogens
- Rise of generative AI as a dual use technology for biodefense and misuse

Enhanced Risks

- Automated bioweapon design with generative models
- Relatively low computing needs to generate dangerous agents
- Challenges detecting artificially generated sequences/organisms

Enhanced Response Capabilities

- Generative models for vaccine and therapeutic design
- High-throughput testing of countermeasures with synthetic data
- AI for early detection of emergent pathogens and outbreaks

Recommendations

- Increased oversight for generative model development/release
- Expanding biosecurity legislation and regulations
- Fostering open research and global cooperation

Outlook

- Trajectory toward increasingly powerful generative biological capabilities
- Need for preventative ethics research and guidance
- Maintaining public trust and avoiding overreaction

Conclusion

- Balancing generative AI's benefits and risks in biology
- Importance of thoughtful governance and responsible innovation
- Staying ahead of the curve on biosecurity

Policy Responses to Generative AI in Biology

Introduction

- Background on rise of powerful generative models for biology
- Overview of risks like bioweapons, environmental damage
- Need for governance to ensure responsible development

Self-Governance by Developers

- Voluntary guidelines on ethical AI by corporations
- Limiting access to certain capabilities like human editing
- Issues with self-regulation and transparency

Governmental Regulations

- New biosecurity regulations on certain AI technologies
- Restrictions on use of synthetic biology IP
- Challenges with fast pace of technology change

International Governance

- Proposals for global observatory to monitor risks
- Treaties restricting development of bioweapons
- Difficulty achieving consensus and compliance

Public Deliberation and Ethics

- Activities to involve broader stakeholders
- Gathering public attitudes on acceptable uses of generative bio AI
- Informing policy with deliberative democracy

Outlook

- Likely increase in debate and policy activity in this space
- Balancing innovation and security will be a key challenge
- Importance of thoughtful multidisciplinary discourse

Conclusion

- No easy policy solutions, but inaction also carries risks
- Policy should enable innovation but promote responsible use
- This will require sustained public deliberation and coordination

References

1. **Computers in the study of evolution**
JL Crosby
Science Progress (1967) https://pubmed.ncbi.nlm.nih.gov/4859964

2. **The chemical basis of morphogenesis**
Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences (1952-08-14) https://royalsocietypublishing.org/doi/10.1098/rstb.1952.0012
DOI: 10.1098/rstb.1952.0012

3. **Scientific uses of the MANIAC**
HL Anderson
Journal of Statistical Physics (1986-06-01) https://doi.org/10.1007/BF02628301
DOI: 10.1007/bf02628301

4. **A three-dimensional model of the myoglobin molecule obtained by x-ray analysis**
JC Kendrew, G Bodo, HM Dintzis, RG Parrish, H Wyckoff, DC Phillips
Nature (1958-03-08) https://pubmed.ncbi.nlm.nih.gov/13517261
DOI: 10.1038/181662a0

5. **Structure of myoglobin: A three-dimensional Fourier synthesis at 2 A. resolution**
JC Kendrew, RE Dickerson, BE Strandberg, RG Hart, DR Davies, DC Phillips, VC Shore
Nature (1960-02-13) https://pubmed.ncbi.nlm.nih.gov/18990802
DOI: 10.1038/185422a0

6. **The amino-acid sequence in the glycyl chain of insulin. 1. The identification of lower peptides from partial hydrolysates**
F Sanger, EOP Thompson
Biochemical Journal (1953-02) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1198157/

7. **The amino-acid sequence in the glycyl chain of insulin. 2. The investigation of peptides from enzymic hydrolysates**
F Sanger, EOP Thompson
Biochemical Journal (1953-02) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1198158/

8. **A method for the determination of amino acid sequence in peptides**
P Edman
Archives of Biochemistry (1949-07) https://pubmed.ncbi.nlm.nih.gov/18134557

9. **The origins of bioinformatics**
JB Hagen
Nature Reviews. Genetics (2000-12) https://pubmed.ncbi.nlm.nih.gov/11252753
DOI: 10.1038/35042090

10. **Comprotein: a computer program to aid primary protein structure determination**
Margaret Oakley Dayhoff, Robert S Ledley
Proceedings of the December 4-6, 1962, fall joint computer conference on - AFIPS '62 (Fall) (1962) http://portal.acm.org/citation.cfm?doid=1461518.1461546
DOI: 10.1145/1461518.1461546

11. https://febs.onlinelibrary.wiley.com/toc/14321033/5/2

12. **Collecting, comparing, and computing sequences: the making of Margaret O. Dayhoff's Atlas of Protein Sequence and Structure, 1954-1965**
Bruno J Strasser
Journal of the History of Biology (2010) https://pubmed.ncbi.nlm.nih.gov/20665074
DOI: 10.1007/s10739-009-9221-0

13. **A general method applicable to the search for similarities in the amino acid sequence of two proteins**
SB Needleman, CD Wunsch
Journal of Molecular Biology (1970-03) https://pubmed.ncbi.nlm.nih.gov/5420325
DOI: 10.1016/0022-2836(70)90057-4

14. **Progressive sequence alignment as a prerequisite to correct phylogenetic trees**
DF Feng, RF Doolittle
Journal of Molecular Evolution (1987) https://pubmed.ncbi.nlm.nih.gov/3118049
DOI: 10.1007/bf02603120

15. **CLUSTAL: a package for performing multiple sequence alignment on a microcomputer**
DG Higgins, PM Sharp
Gene (1988-12-15) https://pubmed.ncbi.nlm.nih.gov/3243435
DOI: 10.1016/0378-1119(88)90330-7

16. **Clustal Omega, accurate alignment of very large numbers of sequences**
Fabian Sievers, Desmond G Higgins
Methods in Molecular Biology (Clifton, N.J.) (2014) https://pubmed.ncbi.nlm.nih.gov/24170397
DOI: 10.1007/978-1-62703-646-7_6

17. **A new method for sequencing DNA**
AM Maxam, W Gilbert
Proceedings of the National Academy of Sciences of the United States of America (1977-02) https://pubmed.ncbi.nlm.nih.gov/265521
DOI: 10.1073/pnas.74.2.560

18. **Summary statement of the Asilomar conference on recombinant DNA molecules.**
P Berg, D Baltimore, S Brenner, RO Roblin, MF Singer
Proceedings of the National Academy of Sciences of the United States of America (1975-06) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC432675/

19. **A strategy of DNA sequencing employing computer programs.**
R Staden
Nucleic Acids Research (1979-06-11) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC327874/

20. **Evolutionary trees from DNA sequences: a maximum likelihood approach**
J Felsenstein
Journal of Molecular Evolution (1981) https://pubmed.ncbi.nlm.nih.gov/7288891
DOI: 10.1007/bf01734359

21. **Probability distribution of molecular evolutionary trees: a new method of phylogenetic inference**
B Rannala, Z Yang
Journal of Molecular Evolution (1996-09) https://pubmed.ncbi.nlm.nih.gov/8703097
DOI: 10.1007/bf02338839

22. **A biologist's guide to Bayesian phylogenetic analysis**
Fabrícia F Nascimento, Mario Dos Reis, Ziheng Yang
Nature Ecology & Evolution (2017-10) https://pubmed.ncbi.nlm.nih.gov/28983516

DOI: 10.1038/s41559-017-0280-x

23. **The Nobel Prize in Chemistry 1993**
NobelPrize.org
https://www.nobelprize.org/prizes/chemistry/1993/mullis/lecture/

24. **A comprehensive set of sequence analysis programs for the VAX**
J Devereux, P Haeberli, O Smithies
Nucleic Acids Research (1984-01-11) https://pubmed.ncbi.nlm.nih.gov/6546423
DOI: 10.1093/nar/12.1part1.387

25. **DNASTAR's Lasergene Sequence Analysis Software**
Timothy G Burland
Bioinformatics Methods and Protocols (1999) https://doi.org/10.1385/1-59259-192-2:71
ISBN: 9781592591923

26. **Apple II PASCAL programs for molecular biologists**
B Malthiery, B Bellon, D Giorgi, B Jacq
Nucleic Acids Research (1984-01-11) https://pubmed.ncbi.nlm.nih.gov/6320099
DOI: 10.1093/nar/12.1part2.569

27. **The GNU Manifesto - GNU Project - Free Software Foundation**
https://www.gnu.org/gnu/manifesto.en.html

28. https://www.researchgate.net/publication/221307757_The_Free_Software_Movement_and_the_GNULinux_Operating_System

29. **Whole-genome random sequencing and assembly of Haemophilus influenzae Rd**
RD Fleischmann, MD Adams, O White, RA Clayton, EF Kirkness, AR Kerlavage, CJ Bult, JF Tomb, BA Dougherty, JM Merrick
Science (New York, N.Y.) (1995-07-28) https://pubmed.ncbi.nlm.nih.gov/7542800
DOI: 10.1126/science.7542800

30. **The sequence of the human genome**
JC Venter, MD Adams, EW Myers, PW Li, RJ Mural, GG Sutton, HO Smith, M Yandell, CA Evans, RA Holt, … X Zhu
Science (New York, N.Y.) (2001-02-16) https://pubmed.ncbi.nlm.nih.gov/11181995
DOI: 10.1126/science.1058040

31. **Consed: a graphical tool for sequence finishing**
D Gordon, C Abajian, P Green
Genome Research (1998-03) https://pubmed.ncbi.nlm.nih.gov/9521923
DOI: 10.1101/gr.8.3.195

32. **A whole-genome assembly of Drosophila**
EW Myers, GG Sutton, AL Delcher, IM Dew, DP Fasulo, MJ Flanigan, SA Kravitz, CM Mobarry, KH Reinert, KA Remington, … JC Venter
Science (New York, N.Y.) (2000-03-24) https://pubmed.ncbi.nlm.nih.gov/10731133
DOI: 10.1126/science.287.5461.2196

33. **The EMBL data library**
CM Rice, R Fuchs, DG Higgins, PJ Stoehr, GN Cameron
Nucleic Acids Research (1993-07-01) https://pubmed.ncbi.nlm.nih.gov/8332519
DOI: 10.1093/nar/21.13.2967

34. **GenBank**

Dennis A Benson, Mark Cavanaugh, Karen Clark, Ilene Karsch-Mizrachi, David J Lipman, James Ostell, Eric W Sayers
Nucleic Acids Research (2013-01) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3531190/
DOI: 10.1093/nar/gks1195

35. **BLAST: at the core of a powerful and diverse set of sequence analysis tools**
Scott McGinnis, Thomas L Madden
Nucleic Acids Research (2004-07-01) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC441573/
DOI: 10.1093/nar/gkh435

36. **Genomes OnLine Database (GOLD): a monitor of genome projects world-wide**
Axel Bernal, Uy Ear, Nikos Kyrpides
Nucleic Acids Research (2001-01-01) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC29859/

37. **PubMed**
PubMed
https://pubmed.ncbi.nlm.nih.gov/

38. https://academic.oup.com/nar/article/26/1/94/2379498

39. Abramson

40. **The BLEND system Programme for the study of some 'electronic journals'∗**
B Shackel
Ergonomics (1982-04) http://www.tandfonline.com/doi/abs/10.1080/00140138208924954
DOI: 10.1080/00140138208924954

41. **arXiv.org e-Print archive** https://arxiv.org/

42. **bioRxiv.org - the preprint server for Biology** https://www.biorxiv.org/

43. **Configurations of Polypeptide Chains With Favored Orientations Around Single Bonds: Two New Pleated Sheets**
L Pauling, RB Corey
Proceedings of the National Academy of Sciences of the United States of America (1951-11)
https://pubmed.ncbi.nlm.nih.gov/16578412
DOI: 10.1073/pnas.37.11.729

44. **Sixty-five years of the long march in protein secondary structure prediction: the final stretch?**
Yuedong Yang, Jianzhao Gao, Jihua Wang, Rhys Heffernan, Jack Hanson, Kuldip Paliwal, Yaoqi Zhou
Briefings in Bioinformatics (2018-05-01) https://pubmed.ncbi.nlm.nih.gov/28040746
DOI: 10.1093/bib/bbw129

45. **Computational methods for protein structure prediction and modeling**
New York, N.Y. : Springer
(2007) http://archive.org/details/computationalmet0000unse_u4q5
ISBN: 9780387333212

46. **Molecular dynamics simulations: advances and applications**
Adam Hospital, Josep Ramon Goñi, Modesto Orozco, Josep L Gelpí
Advances and Applications in Bioinformatics and Chemistry : AABC (2015-11-19)
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4655909/
DOI: 10.2147/aabc.s70333

47. **To Milliseconds and Beyond: Challenges in the Simulation of Protein Folding**
Thomas J Lane, Diwakar Shukla, Kyle A Beauchamp, Vijay S Pande
Current opinion in structural biology (2013-02)
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3673555/
DOI: 10.1016/j.sbi.2012.11.002

48. **Next-Generation Sequencing and Its Application: Empowering in Public Health Beyond Reality**
Nidhi Gupta, Vijay K Verma
Microbial Technology for the Welfare of Society (2019-09-13)
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7122948/
DOI: 10.1007/978-981-13-8844-6_15

49. **BOINC: A Platform for Volunteer Computing**
David P Anderson
Journal of Grid Computing (2020-03-01) https://doi.org/10.1007/s10723-019-09497-9
DOI: 10.1007/s10723-019-09497-9

50. **Structural proteomics by NMR spectroscopy**
Joon Shin, Woonghee Lee, Weontae Lee
Expert Review of Proteomics (2008-08) https://pubmed.ncbi.nlm.nih.gov/18761469
DOI: 10.1586/14789450.5.4.589

51. **A whole-cell computational model predicts phenotype from genotype**
Jonathan R Karr, Jayodita C Sanghvi, Derek N Macklin, Miriam V Gutschow, Jared M Jacobs, Benjamin Bolival, Nacyra Assad-Garcia, John I Glass, Markus W Covert
Cell (2012-07-20) https://pubmed.ncbi.nlm.nih.gov/22817898
DOI: 10.1016/j.cell.2012.05.044

52. **Using deep learning to model the hierarchical structure and function of a cell**
Jianzhu Ma, Michael Ku Yu, Samson Fong, Keiichiro Ono, Eric Sage, Barry Demchak, Roded Sharan, Trey Ideker
Nature Methods (2018-04) https://www.nature.com/articles/nmeth.4627
DOI: 10.1038/nmeth.4627

53. **Why Build Whole-Cell Models?**
Javier Carrera, Markus W Covert
Trends in Cell Biology (2015-12) https://pubmed.ncbi.nlm.nih.gov/26471224
DOI: 10.1016/j.tcb.2015.09.004

54. **The future of whole-cell modeling**
Derek N Macklin, Nicholas A Ruggero, Markus W Covert
Current Opinion in Biotechnology (2014-08) https://pubmed.ncbi.nlm.nih.gov/24556244
DOI: 10.1016/j.copbio.2014.01.012

55. **Computer-Aided Whole-Cell Design: Taking a Holistic Approach by Integrating Synthetic With Systems Biology**
Lucia Marucci, Matteo Barberis, Jonathan Karr, Oliver Ray, Paul R Race, Miguel de Souza Andrade, Claire Grierson, Stefan Andreas Hoffmann, Sophie Landon, Elibio Rech, … Christopher Woods
Frontiers in Bioengineering and Biotechnology (2020-08-07)
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7426639/
DOI: 10.3389/fbioe.2020.00942

56. **Accelerated discovery via a whole-cell model**

Jayodita C Sanghvi, Sergi Regot, Silvia Carrasco, Jonathan R Karr, Miriam V Gutschow, Benjamin Bolival, Markus W Covert
Nature methods (2013-12) https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3856890/
DOI: 10.1038/nmeth.2724

57. **A blueprint for human whole-cell modeling**
Balázs Szigeti, Yosef D Roth, John AP Sekar, Arthur P Goldberg, Saahith C Pochiraju, Jonathan R Karr
Current opinion in systems biology (2018-02)
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5966287/
DOI: 10.1016/j.coisb.2017.10.005

58. https://www.researchgate.net/publication/30874496_Artificial_Intelligence_A_Modern_Approach

59. https://www.researchgate.net/publication/223020409_Chen_CM_Intelligent_Web-based_Learning_System_with_Personalized_Learning_Path_Guidance_Computers_Education_51_2_787-814

60. **Expert systems: An overview | IEEE Journals & Magazine | IEEE Xplore**
https://ieeexplore.ieee.org/document/1145205

61. **On Interface Requirements for Expert Systems**
R Wexelblat
The AI Magazine (1989) https://www.semanticscholar.org/paper/On-Interface-Requirements-for-Expert-Systems-Wexelblat/291bffa7fec4fafff62462d015dd86c466273d4c

62. https://stacks.stanford.edu/file/druid:pj337tr4694/pj337tr4694.pdf

63. **MYCIN: a knowledge-based consultation program for infectious disease diagnosis**
William van Melle
International Journal of Man-Machine Studies (1978-05-01)
https://www.sciencedirect.com/science/article/pii/S0020737378800492
DOI: 10.1016/s0020-7373(78)80049-2

64. https://www.researchgate.net/publication/235028224_The_Applicability_and_Limitations_of_Expert_System_Shells

65. **Artificial intelligence, machine learning and deep learning: Potential resources for the infection clinician**
Anastasia A Theodosiou, Robert C Read
The Journal of Infection (2023-10) https://pubmed.ncbi.nlm.nih.gov/37468046
DOI: 10.1016/j.jinf.2023.07.006

66. **A survey on semi-supervised learning**
Jesper E van Engelen, Holger H Hoos
Machine Learning (2020-02-01) https://doi.org/10.1007/s10994-019-05855-6
DOI: 10.1007/s10994-019-05855-6

67. **Unsupervised K-Means Clustering Algorithm | IEEE Journals & Magazine | IEEE Xplore**
https://ieeexplore.ieee.org/document/9072123

68. **Efficient Training Management for Mobile Crowd-Machine Learning: A Deep Reinforcement Learning Approach | IEEE Journals & Magazine | IEEE Xplore**
https://ieeexplore.ieee.org/document/8716527

69. **Machine learning in construction: From shallow to deep learning**

Yayin Xu, Ying Zhou, Przemyslaw Sekula, Lieyun Ding
Developments in the Built Environment (2021-05-01)
https://www.sciencedirect.com/science/article/pii/S2666165921000041
DOI: 10.1016/j.dibe.2021.100045

70. **An Introduction to Convolutional Neural Networks**
Keiron O'Shea, Ryan Nash
arXiv (2015-12-02) http://arxiv.org/abs/1511.08458
DOI: 10.48550/arxiv.1511.08458

71. **Recurrent Neural Networks (RNNs): A gentle Introduction and Overview**
Robin M Schmidt
arXiv (2019-11-23) http://arxiv.org/abs/1912.05911
DOI: 10.48550/arxiv.1912.05911

72. **Distributed representations, simple recurrent networks, and grammatical structure**
Jeffrey L Elman
Machine Learning (1991-09-01) https://doi.org/10.1007/BF00114844
DOI: 10.1007/bf00114844

73. **Autoencoders**
Dor Bank, Noam Koenigstein, Raja Giryes
arXiv (2021-04-03) http://arxiv.org/abs/2003.05991
DOI: 10.48550/arxiv.2003.05991

74. **Generative Adversarial Networks**
Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio
arXiv (2014-06-10) http://arxiv.org/abs/1406.2661
DOI: 10.48550/arxiv.1406.2661

75. **Natural language processing: state of the art, current trends and challenges**
Diksha Khurana, Aditya Koli, Kiran Khatter, Sukhdev Singh
Multimedia Tools and Applications (2023) https://pubmed.ncbi.nlm.nih.gov/35855771
DOI: 10.1007/s11042-022-13428-4

76. **Generative AI: A systematic review using topic modelling techniques**
Priyanka Gupta, Bosheng Ding, Chong Guan, Ding Ding
Data and Information Management (2024-06-01)
https://www.sciencedirect.com/science/article/pii/S2543925124000020
DOI: 10.1016/j.dim.2024.100066

77. **Auto-Encoding Variational Bayes**
Diederik P Kingma, Max Welling
arXiv (2022-12-10) http://arxiv.org/abs/1312.6114
DOI: 10.48550/arxiv.1312.6114

78. https://www.researchgate.net/publication/377955158_Autoencoders_and_their_applications_in_machine_learning_a_survey

79. **Unsupervised deep learning with variational autoencoders applied to breast tumor genome-wide DNA methylation data with biologic feature extraction**
Alexander J Titus, Owen M Wilkins, Carly A Bobak, Brock C Christensen
bioRxiv (2018-11-07) https://www.biorxiv.org/content/10.1101/433763v5
DOI: 10.1101/433763

80. https://www.researchgate.net/publication/322870935_A_New_Dimension_of_Breast_Cancer_Epigenetics_-_Applications_of_Variational_Autoencoders_with_DNA_Methylation

81. **A Style-Based Generator Architecture for Generative Adversarial Networks**
Tero Karras, Samuli Laine, Timo Aila
arXiv (2019-03-29) http://arxiv.org/abs/1812.04948
DOI: 10.48550/arxiv.1812.04948

82. **Denoising Diffusion Probabilistic Models**
Jonathan Ho, Ajay Jain, Pieter Abbeel
arXiv (2020-12-16) http://arxiv.org/abs/2006.11239
DOI: 10.48550/arxiv.2006.11239

83. **High-Resolution Image Synthesis with Latent Diffusion Models**
Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, Björn Ommer
arXiv (2022-04-13) http://arxiv.org/abs/2112.10752
DOI: 10.48550/arxiv.2112.10752

84. **Attention Is All You Need**
Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, Illia Polosukhin
arXiv (2023-08-01) http://arxiv.org/abs/1706.03762
DOI: 10.48550/arxiv.1706.03762

85. https://openai.com/index/chatgpt

86. **Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer**
Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu
arXiv (2023-09-19) http://arxiv.org/abs/1910.10683
DOI: 10.48550/arxiv.1910.10683

87. **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**
Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova
arXiv (2019-05-24) http://arxiv.org/abs/1810.04805
DOI: 10.48550/arxiv.1810.04805

88. **Language Models are Few-Shot Learners**
Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, … Dario Amodei
arXiv (2020-07-22) http://arxiv.org/abs/2005.14165
DOI: 10.48550/arxiv.2005.14165

89. **A Survey of Large Language Models in Medicine: Progress, Application, and Challenge**
Hongjian Zhou, Fenglin Liu, Boyang Gu, Xinyu Zou, Jinfa Huang, Jinge Wu, Yiru Li, Sam S Chen, Peilin Zhou, Junling Liu, … David A Clifton
arXiv (2024-05-15) http://arxiv.org/abs/2311.05112
DOI: 10.48550/arxiv.2311.05112

90. **HyenaDNA: Long-Range Genomic Sequence Modeling at Single Nucleotide Resolution**
Eric Nguyen, Michael Poli, Marjan Faizi, Armin Thomas, Callum Birch-Sykes, Michael Wornow, Aman Patel, Clayton Rabideau, Stefano Massaroli, Yoshua Bengio, … Chris Ré
arXiv (2023) https://doi.org/gs3v5j
DOI: 10.48550/arxiv.2306.15794