

A brief history of bioinformatics

Jeff Gauthier, Antony T. Vincent, Steve J. Charette and Nicolas Derome

Corresponding author: Jeff Gauthier, Institut de Biologie Intégrative et des Systèmes, 1030 avenue de la Médecine, Université Laval, Québec City, QC G1V0A6, Canada. Tel.: 1 (418) 656-2131 # 6785; Fax: 1 (418) 656-7176; E-mail: jeff.gauthier.1@ulaval.ca

Abstract

It is easy for today's students and researchers to believe that modern bioinformatics emerged recently to assist next-generation sequencing data analysis. However, the very beginnings of bioinformatics occurred more than 50 years ago, when desktop computers were still a hypothesis and DNA could not yet be sequenced. The foundations of bioinformatics were laid in the early 1960s with the application of computational methods to protein sequence analysis (notably, *de novo* sequence assembly, biological sequence databases and substitution models). Later on, DNA analysis also emerged due to parallel advances in (i) molecular biology methods, which allowed easier manipulation of DNA, as well as its sequencing, and (ii) computer science, which saw the rise of increasingly miniaturized and more powerful computers, as well as novel software better suited to handle bioinformatics tasks. In the 1990s through the 2000s, major improvements in sequencing technology, along with reduced costs, gave rise to an exponential increase of data. The arrival of 'Big Data' has laid out new challenges in terms of data mining and management, calling for more expertise from computer science into the field. Coupled with an ever-increasing amount of bioinformatics tools, biological Big Data had (and continues to have) profound implications on the predictive power and reproducibility of bioinformatics results. To overcome this issue, universities are now fully integrating this discipline into the curriculum of biology students. Recent subdisciplines such as synthetic biology, systems biology and whole-cell modeling have emerged from the ever-increasing complementarity between computer science and biology.

Key words: bioinformatics; origin of bioinformatics; genomics; structural bioinformatics; Big Data; future of bioinformatics

Introduction

Computers and specialized software have become an essential part of the biologist's toolkit. Either for routine DNA or protein sequence analysis or to parse meaningful information in massive gigabyte-sized biological data sets, virtually all modern research projects in biology require, to some extent, the use of computers. This is especially true since the advent of next-generation sequencing (NGS) that fundamentally changed the ways of population genetics, quantitative genetics, molecular systematics, microbial ecology and many more research fields.

In this context, it is easy for today's students and researchers to believe that modern bioinformatics are relatively recent, coming to the rescue of NGS data analysis. However, the very beginnings of bioinformatics occurred more than 50 years ago, when desktop computers were still a hypothesis and DNA could not yet be sequenced. Here we present an integrative timeline

of key events in bioinformatics and related fields during the past half-century, as well as some background on parallel advances in molecular biology and computer science, and some reflections on the future of bioinformatics. We hope this review helps the reader to understand what made bioinformatics become the major driving force in biology that it is today.

1950–1970: The origins

It did not start with DNA analysis

In the early 1950s, not much was known about deoxyribonucleic acid (DNA). Its status as the carrier molecule of genetic information was still controversial at that time. Avery, MacLeod and McCarty (1944) showed that the uptake of pure DNA from a virulent bacterial strain could confer virulence to a nonvirulent strain [1], but their results were not immediately accepted by

Jeff Gauthier is a PhD candidate in biology at Université Laval, where he studies the role of host-microbiota interactions in salmonid fish diseases.

Antony T. Vincent is a Postdoctoral Fellow at INRS-Institut Armand-Frappier, Laval, Canada, where he studies bacterial evolution.

Steve J. Charette is a Full Professor at Université Laval where he investigates bacterial pathogenesis and other host-microbe interactions.

Nicolas Derome is a Full Professor at Université Laval where he investigates host-microbiota interactions and the evolution of microbial communities.

Submitted: 26 February 2018; Received (in revised form): 22 June 2018

© The Author(s) 2018. Published by Oxford University Press. All rights reserved. For permissions, please email: journals.permissions@oup.com

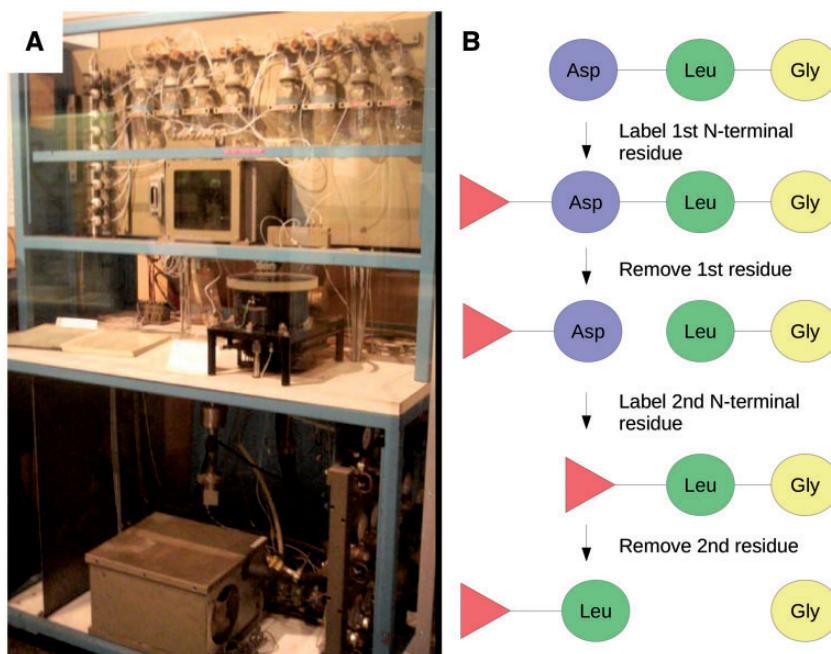


Figure 1. Automated Edman peptide sequencing. (A) One of the first automated peptide sequencers, designed by William J. Dreyer. (B) Edman sequencing: the first N-terminal amino acid of a peptide chain is labeled with phenylisothiocyanate (PITC, red triangle), and then cleaved by lowering the pH. By repeating this process, one can determine a peptide sequence, one N-terminal amino acid at a time.

the scientific community. Many thought that proteins were the carriers of genetic information [2]. The role of DNA as a genetic information encoding molecule was validated in 1952 by Hershey and Chase when they proved beyond reasonable doubt that it was DNA, not protein, that was taken up and transmitted by bacterial cells infected by a bacteriophage [3].

Despite the knowledge of its major role, not much was known about the arrangement of the DNA molecule. All we knew was that pairs of its monomers (i.e. nucleotides) were in equimolar proportions [4]. In other words, there is as much adenosine as there is thymidine, and there is as much guanine as there is cytidine. It was in 1953 that the double-helix structure of DNA was finally solved by Watson, Crick and Franklin [5]. Despite this breakthrough, it would take 13 more years before deciphering the genetic code [6] and 25 more years before the first DNA sequencing methods became available [7, 8]. Consequently, the use of bioinformatics in DNA analysis lagged nearly two decades behind the analysis of proteins, whose chemical nature was already better understood than DNA.

Protein analysis was the starting point

In the late 1950s, in addition to major advances in determination of protein structures through crystallography [9], the first sequence (i.e. amino acid chain arrangement) of a protein, insulin, was published [10, 11]. This major leap settled the debate about the polypeptide chain arrangement of proteins [12]. Furthermore, it encouraged the development of more efficient methods for obtaining protein sequences. The Edman degradation method [13] emerged as a simple method that allowed protein sequencing, one amino acid at a time starting from the N-terminus. Coupled with automation (Figure 1), more than 15 different protein families were sequenced over the following 10 years [12].

A major issue with Edman sequencing was obtaining large protein sequences. Edman sequencing works through one-by-one cleavage of N-terminal amino acid residues with

phenylisothiocyanate [13]. However, the yield of this reaction is never complete. Because of this, a theoretical maximum of 50–60 amino acids can be sequenced in a single Edman reaction [14]. Larger proteins must be cleaved into smaller fragments, which are then separated and individually sequenced.

The issue was not sequencing a protein in itself but rather assembling the whole protein sequence from hundreds of small Edman peptide sequences. For large proteins made of several hundreds (if not thousands) of residues, getting back the final sequence was cumbersome. In the early 1960s, one of the first known bioinformatics software was developed to solve this problem.

Dayhoff: the first bioinformatician

Margaret Dayhoff (1925–1983) was an American physical chemist who pioneered the application of computational methods to the field of biochemistry. Dayhoff's contribution to this field is so important that David J. Lipman, former director of the National Center for Biotechnology Information (NCBI), called her 'the mother and father of bioinformatics' [15].

Dayhoff had extensively used computational methods for her PhD thesis in electrochemistry [16] and saw the potential of computers in the fields of biology and medicine. In 1960, she became Associate Director of the National Biomedical Resource Foundation. There, she began to work with Robert S. Ledley, a physicist who also sought to bring computational resources to biomedical problems [17, 18]. From 1958 to 1962, both combined their expertise and developed COMPROTEIN, 'a complete computer program for the IBM 7090' designed to determine protein primary structure using Edman peptide sequencing data [19]. This software, entirely coded in FORTRAN on punch-cards, is the first occurrence of what we would call today a *de novo* sequence assembler (Figure 2).

In the COMPROTEIN software, input and output amino acid sequences were represented in three-letter abbreviations (e.g. Lys for lysine, Ser for serine). In an effort to simplify the

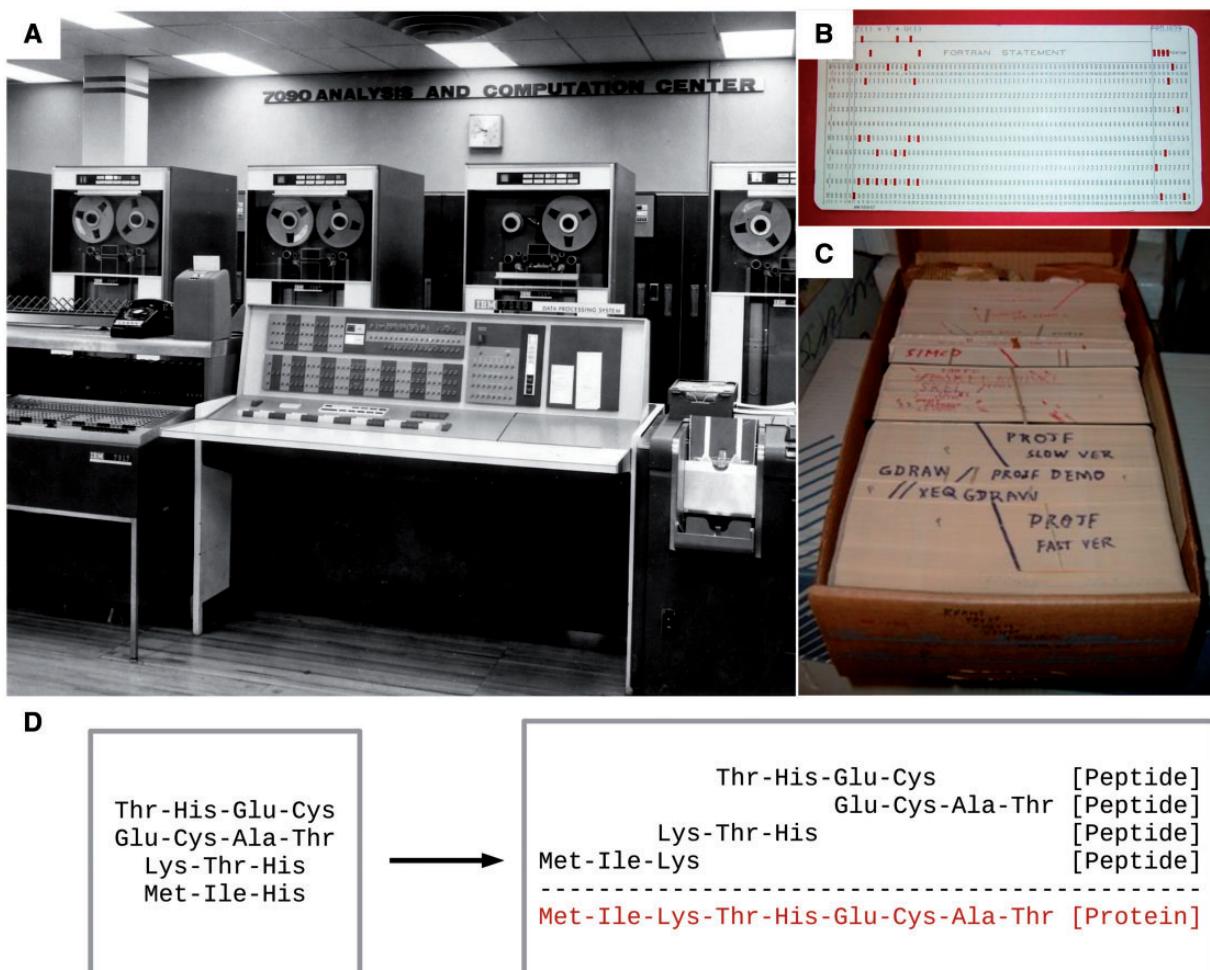


Figure 2. COMPROTEIN, the first bioinformatics software. (A) An IBM 7090 mainframe, for which COMPROTEIN was made to run. (B) A punch card containing one line of FORTRAN code (the language COMPROTEIN was written with). (C) An entire program's source code in punch cards. (D) A simplified overview of COMPROTEIN's input (i.e. Edman peptide sequences) and output (a consensus protein sequence).

handling of protein sequence data, Dayhoff later developed the one-letter amino acid code that is still in use today [20]. This one-letter code was first used in Dayhoff and Eck's 1965 *Atlas of Protein Sequence and Structure* [21], the first ever biological sequence database. The first edition of the *Atlas* contained 65 protein sequences, most of which were interspecific variants of a handful of proteins. For this reason, the first *Atlas* happened to be an ideal data set for two researchers who hypothesized that protein sequences reflect the evolutionary history of species.

The computer-assisted genealogy of life

Although much of the pre-1960s research in biochemistry focused on the mechanistic modeling of enzymes [22], Emile Zuckerkandl and Linus Pauling departed from this paradigm by investigating biomolecular sequences as 'carriers of information'. Just as words are strings of letters whose specific arrangement convey meaning, the molecular function (i.e. meaning) of a protein results from how its amino acids are arranged to form a 'word' [23]. Knowing that words and languages evolve by inheritance of subtle changes over time [24], could protein sequences evolve through a similar mechanism? Could these inherited changes allow biologists to reconstitute the evolutionary history of those proteins, and in the same

process, reconstitute the sequence of their 'ancestors'? Zuckerkandl and Pauling in 1963 coined the term 'Paleogenetics' to introduce this novel branch of evolutionary biology [25].

Both observed that orthologous proteins from vertebrate organisms, such as hemoglobin, showed a degree of similarity too high over long evolutionary time to be the result of either chance or convergent evolution (*Ibid*). The concept of orthology itself was defined in 1970 by Walter M. Fitch to describe homology that resulted from a speciation event [26]. Furthermore, the amount of differences in orthologs from different species seemed proportional to the evolutionary divergence between those species. For instance, they observed that human hemoglobin showed higher conservation with chimpanzee (*Pan troglodytes*) hemoglobin than with mouse (*Mus musculus*) hemoglobin. This sequence identity gradient correlated with divergence estimates derived from the fossil record (Figure 3).

In light of these observations, Zuckerkandl and Pauling hypothesized that orthologous proteins evolved through divergence from a common ancestor. Consequently, by comparing the sequence of hemoglobin in currently extant organisms, it became possible to predict the 'ancestral sequences' of hemoglobin and, in the process, its evolutionary history up to its current forms.

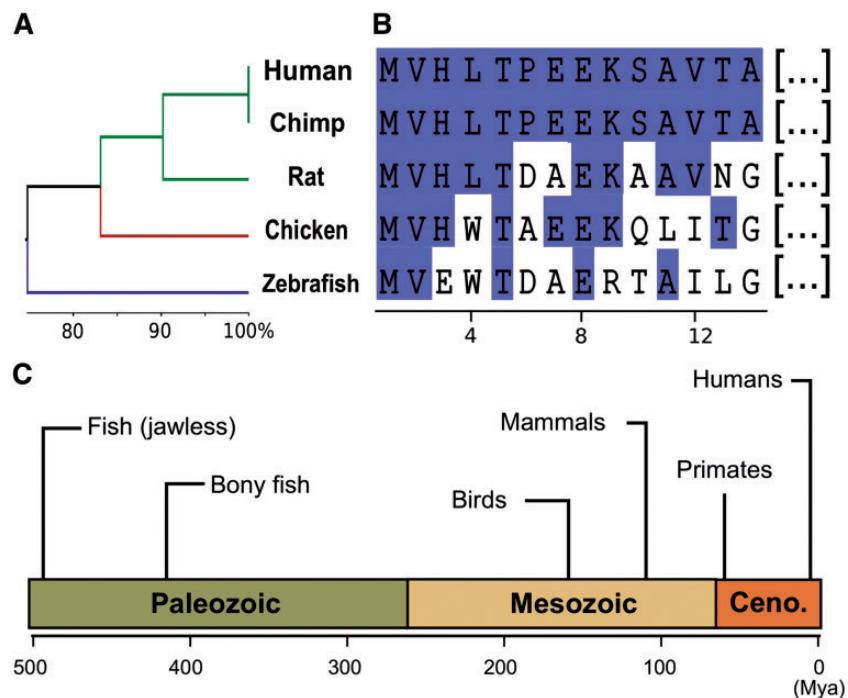


Figure 3. Sequence dissimilarity between orthologous proteins from model organisms correlates with their evolutionary history as evidenced by the fossil record. (A) Average distance tree of hemoglobin subunit beta-1 (HBB-1) from human (*Homo sapiens*), chimpanzee (*Pan troglodytes*), rat (*Rattus norvegicus*), chicken (*Gallus gallus*) and zebrafish (*Danio rerio*). (B) Alignment view of the first 14 amino acid residues of HBB-1 compared in (A) (residues highlighted in blue are identical to the human HBB-1 sequence). (C) Timeline of earliest fossils found for different aquatic and terrestrial animals.

However, several conceptual and computational problems were to be solved, notably, judging the ‘evolutionary value’ of substitutions between protein sequences. Moreover, the absence of reproducible algorithms for aligning protein sequences was also an issue. In the first sequence-based phylogenetic studies, the proteins that were investigated (mostly homologs from different mammal species) were so closely related that visual comparison was sufficient to assess homology between sites [27]. For proteins sharing a more distant ancestor, or proteins of unequal sequence length, this strategy could either be unpractical or lead to erroneous results.

This issue was solved in great part in 1970 by Needleman and Wunsch [28], who developed the first dynamic programming algorithm for pairwise protein sequence alignments (Figure 4). Despite this major advance, it was not until the early 1980s that the first multiple sequence alignment (MSA) algorithms emerged. The first published MSA algorithm was a generalization of the Needleman–Wunsch algorithm, which involved using a scoring matrix whose dimensionality equals the number of sequences [29]. This approach to MSA was computationally impractical, as it required a running time of $O(L^N)$, where L is sequence length and N the amount of sequences [30]. In simple terms, the time required to find an optimal alignment is proportional to sequence length exponentiated by the number of sequences; aligning 10 sequences of 100 residues each would require 10^{18} operations. Aligning tens of proteins of greater length would be impractical with such an algorithm.

The first truly practical approach to MSA was developed by Da-Fei Feng and Russell F. Doolittle in 1987 [31]. Their approach, which they called ‘progressive sequence alignment’, consisted of (i) performing a Needleman–Wunsch alignment for all sequence pairs, (ii) extracting pairwise similarity scores for each pairwise alignment, (iii) using those scores to build a guide tree and then (iv) aligning the two most similar sequences, and then

the next more similar sequence, and so on, according to the guide tree. The popular MSA software CLUSTAL was developed in 1988 as a simplification of the Feng–Doolittle algorithm [32], and is still used and maintained to this present day [33].

A mathematical framework for amino acid substitutions

In 1978, Dayhoff, Schwartz and Orcutt [34] contributed to another bioinformatics milestone by developing the first probabilistic model of amino acid substitutions. This model, completed 8 years after its inception, was based on the observation of 1572 point accepted mutations (PAMs) in the phylogenetic trees of 71 families of proteins sharing above 85% identity. The result was a 20×20 asymmetric substitution matrix (Table 1) that contained probability values based on the observed mutations of each amino acid (i.e. the probability that each amino acid will change in a given small evolutionary interval). Whereas the principle of (i.e. least number of changes) was used before to quantify evolutionary distance in phylogenetic reconstructions, the PAM matrix introduced the of substitutions as the measurement of evolutionary change.

In the meantime, several milestones in molecular biology were setting DNA as the primary source of biological information. After the elucidation of its molecular structure and its role as the carrier of genes, it became quite clear that DNA would provide unprecedented amounts of biological information.

1970–1980: Paradigm shift from protein to DNA analysis

Deciphering of the DNA language: the genetic code

The specifications for any living being (more precisely, its ‘proteins’) are encoded in the specific nucleotide arrangements of the DNA molecule. This view was formalized in Francis Crick’s

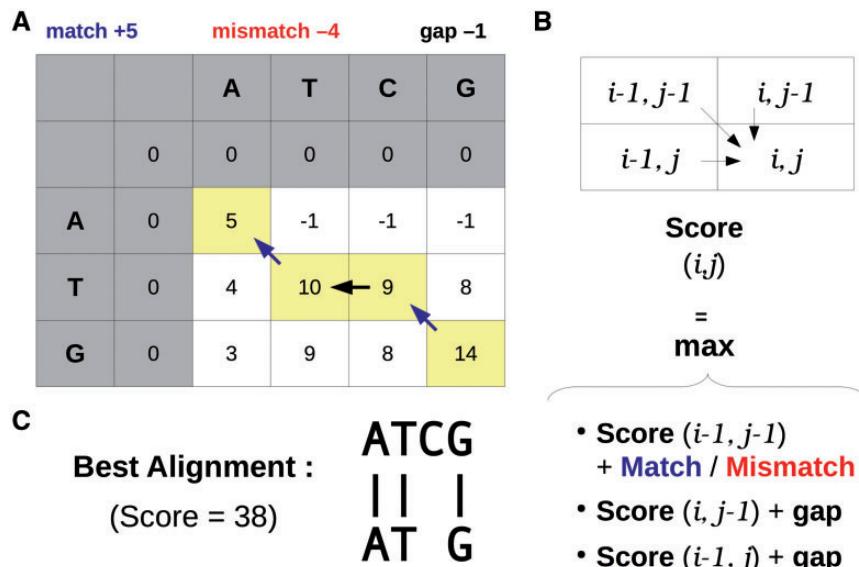


Figure 4. Representation of the Needleman-Wunsch global alignment algorithm. (A) An optimal alignment between two sequences is found by finding the optimal path on a scoring matrix calculated with match and mismatch points (here +5 and -4), and a gap penalty (here -1). (B) Each cell (i, j) of the scoring matrix is calculated with a maximum function, based on the score of neighboring cells. (C) The best alignment between sequences ATCG and ATG, using parameters mentioned in (A). Note: no initial and final gap penalties were defined in this example.

Table 1. An excerpt of the PAM1 amino acid substitution matrix

$10^4 P^a$		Ala	Arg	Asn	Asp	Cys	Gln	...	Val
		A	R	N	D	C	Q	...	V
Ala	A	9867	2	9	10	3	8	...	18
Arg	R	1	9913	1	0	1	10	...	1
Asn	N	4	1	9822	36	0	4	...	1
Asp	D	6	0	42	9859	0	6	...	1
Cys	C	1	1	0	0	9973	0	...	2
Gln	Q	3	9	4	5	0	9876	...	1
...
Val	V	13	2	1	1	3	2	...	9901

^aEach numeric value represents the probability that an amino acid from the i-th column be substituted by an amino acid in the j-th row (multiplied by 10 000).

sequence hypothesis (also called nowadays the ‘Central Dogma’), in which he postulated that RNA sequences, transcribed from DNA, determine the amino acid sequence of the proteins they encode. In turn, the amino acid sequence determines the three-dimensional structure of the protein.

Therefore, if one could figure out how the cell translates the ‘DNA language’ into polypeptide sequences, one could predict the primary structure of any protein produced by an organism by ‘reading its DNA’. By 1968, all of the 64 codons of the genetic code were deciphered [35]; DNA was now ‘readable’, and this groundbreaking achievement called for simple and affordable ways to obtain DNA sequences.

Cost-efficient reading of DNA

The first DNA sequencing method to be widely adopted was the Maxam–Gilbert sequencing method in 1976 [8]. However, its inherent complexity due to extensive use of radioactivity and hazardous chemicals largely prohibited its use in favor of methods developed in Frederick Sanger’s laboratory. Indeed, 25 years after obtaining the first protein sequence [10, 11], Sanger’s team developed the ‘plus and minus’ DNA sequencing method in

1977, the first to rely on primed synthesis with DNA polymerase. The bacteriophage $\Phi X 174$ genome (5386 bp), the first DNA genome ever obtained, was sequenced using this method. Technical modifications to ‘plus and minus’ DNA sequencing led to the common Sanger chain termination method [7], which is still in use today even 40 years after its inception [36].

Being able to obtain DNA sequences from an organism holds many advantages in terms of information throughput. Whereas proteins must be individually purified to be sequenced, the whole genome of an organism can be theoretically derived from a single genomic DNA extract. From this whole-genome DNA sequence, one can predict the primary structure of all proteins expressed by an organism through translation of genes present in the sequence. Though the principle may seem simple, extracting information manually from DNA sequences involves the following:

1. comparisons (e.g. finding homology between sequences from different organisms);
2. calculations (e.g. building a phylogenetic tree of multiple protein orthologs using the PAM1 matrix);
3. and pattern matching (e.g. finding open reading frames in a DNA sequence).

Those tasks are much more efficiently and rapidly performed by computers than by humans. Dayhoff and Eck showed that the computer-assisted analysis of protein sequences yielded more information than mechanistic modeling alone; similarly, the sequence nature of DNA and its remarkable understandability called for a similar approach in its analysis.

The first software dedicated to analyzing Sanger sequencing reads was published by Roger Staden in 1979 [37]. His collection of computer programs could be, respectively, used to (i) search for overlaps between Sanger gel readings; ii) verify, edit and join sequence reads into contigs; and (iii) annotate and manipulate sequence files. The Staden Package was one of the first sequence analysis software to include additional characters (which Staden called ‘uncertainty codes’) to record basecalling uncertainties in a sequence read. This extended DNA alphabet

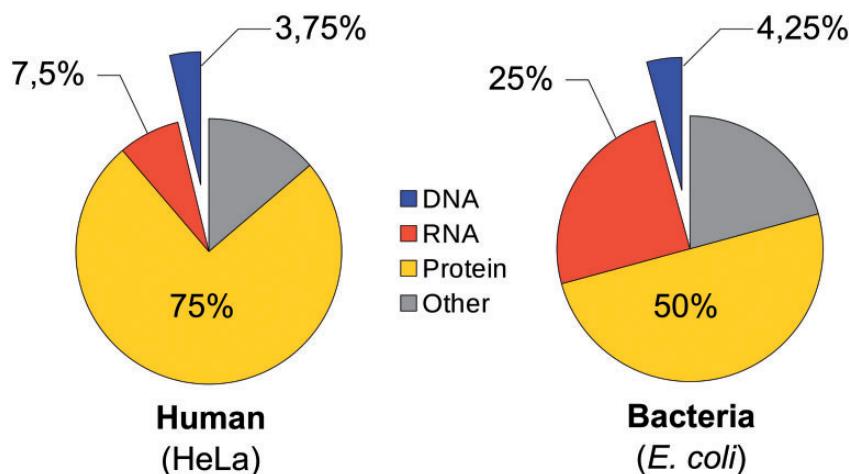


Figure 5. DNA is the least abundant macromolecular cell component that can be sequenced. Percentages (%) represent the abundance of each component relative to total cell dry weight. Data from the Ambion Technical Resource Library.

was one of the precursors of the modern IUBMB (International Union of Biochemistry and Molecular Biology) nomenclature for incompletely specified bases in nucleic acid sequences [38]. The Staden Package is still developed and maintained to this present day (<http://staden.sourceforge.net/>).

Using DNA sequences in phylogenetic inference

Although several concepts related to the notion of phylogeny have been introduced by Ernst Haeckel in 1866 [39], the first molecular phylogenetic trees were reconstructed from protein sequences, and typically assumed maximum parsimony (i.e. the least number of changes) as the main mechanism driving evolutionary change. As stated by Joseph Felsenstein in 1981 [40] about parsimony-based methods,

[They] implicitly assume that change is improbable *a priori* (Felsenstein 1973, 1979). If the amount of change is small over the evolutionary times being considered, parsimony methods will be well-justified statistical methods. Most data involve moderate to large amounts of change, and it is in such cases that parsimony methods can fail.'

Furthermore, the evolution of proteins is driven by how the genes encoding them are modeled by selection, mutation and drift. Therefore, using nucleic acid sequences in phylogenetics added additional information that could not be obtained with amino acid sequences. For instance, synonymous mutations (i.e. a nucleotide substitution that does not modify the amino acids due to the degeneracy of the genetic code) are only detectable when using nucleotide sequences as input. Felsenstein was the first to develop a maximum likelihood (ML) method to infer phylogenetic trees from DNA sequences [40]. Unlike parsimony methods, which reconstruct an evolutionary tree using the least number of changes, ML estimation 'involves finding that evolutionary tree which yields the highest probability of evolving the observed data' [40].

Following Felsenstein's work in molecular phylogeny, several bioinformatics tools using ML were developed and are still widely developed, as are new statistical methods to evaluate the robustness of the nodes. This even inspired in the 1990s, the use of Bayesian statistics in molecular phylogeny [41], which are still commonly used in biology [42].

However, in the second half of the 1970s, several technical limitations had to be overcome to broaden the use of computers

in DNA analysis (not to mention DNA analysis itself). The following decade was pivotal to address these issues.

1980–1990: Parallel advances in biology and computer science

Molecular methods to target and amplify specific genes

Genes, unlike proteins and RNAs, cannot be biochemically fractionated and then individually sequenced, because they all lay contiguously on a handful of DNA molecules per cell. Moreover, genes are usually present in one or few copies per cell. Genes are therefore orders of magnitude less abundant than the products they encode (Figure 5).

This problem was partly solved when Jackson, Symons and Berg (1972) used restriction endonucleases and DNA ligase to cut and insert the circular SV40 viral DNA into lambda DNA, and then transform *Escherichia coli* cells with this construct. As the inserted DNA molecule is replicated in the host organism, it is also amplified as *E. coli* cultures grow, yielding several million copies of a single DNA insert. This experiment pioneered both the isolation and amplification of genes independently from their source organism (for instance, SV40 is a virus that infects primates). However, Berg was so worried about the potential ethical issues (eugenics, warfare and unforeseen biological hazards) that he himself called for a moratorium on the use of recombinant DNA [43]. During the 1975 Asilomar conference, which Berg chaired, a series of guidelines were established, which still live on in the modern practice of genetics.

The second milestone in manipulating DNA was the polymerase chain reaction (PCR), which allows to amplify DNA without cloning procedures. Although the first description of a 'repair synthesis' using DNA polymerase was made in 1971 by Kjell Kleppe et al. [44], the invention of PCR is credited to Kary Mullis [45] because of the substantial optimizations he brought to this method (notably the use of the thermostable Taq polymerase, and development of the thermal cycler). Unlike Kleppe et al., Mullis patented his process, thus gaining much of the recognition for inventing PCR [46].

Both gene cloning and PCR are now commonly used in DNA library preparation, which is critical to obtain sequence data. The emergence of DNA sequencing in the late 1970s, along with enhanced DNA manipulation techniques, has resulted in more



Illustration 1. The DEC PDP-8 Minicomputer, manufactured in 1965. The main processing module (pictured here) weighed 180 pounds and was sold at the introductory price of \$18 500 (~\$140 000 in 2018 dollars). Image: Wikimedia Commons//CC0.

and more available sequence data. In parallel, the 1980s saw increasing access to both computers and bioinformatics software.

Access to computers and specialized software

Before the 1970s, a 'minicomputer' fairly had the dimensions and weight of a small household refrigerator (Illustration 1), excluding terminal and storage units. The size constraint made the acquisition of computers cumbersome for individuals or small workgroups. Even when integrated circuits made their apparition following Intel's 8080 microprocessor in 1974 [47], the first desktop computers lacked user-friendliness to a point that for certain systems (e.g. the Altair 8800), the operating system had to be loaded manually via binary switches on startup [48].

The first wave of ready-to-use microcomputers hit the consumer market in 1977. The three first computers of this wave, namely, the Commodore PET, Apple II and Tandy TRS-80, were small, inexpensive and relatively user-friendly at his time. All three had a built-in BASIC interpreter, which was an easy language for nonprogrammers [49].

The development of microcomputer software for biology came rapidly. In 1984, the University of Wisconsin Genetics Computer Group published the eponymous 'GCG' software suite [50]. The GCG package was a collection of 33 command-line tools to manipulate DNA, RNA or protein sequences. GCG was designed to work on a small-scale mainframe computer (the DEC VAX-11; Illustration 2). This was the first software collection developed for sequence analysis.

Another sequence manipulation suite was developed after GCG in the same year. DNASTAR, which could be run on an CP/M personal computer, has been a popular software suite in



Illustration 2. The DEC VAX-11/780 Minicomputer. From right to left: The computer module, two tape storage units, a monitor and a terminal. The GCG software package was initially designed to run on this computer. Image: Wikimedia Commons//CC0.

the 1980s and 1990s for its capabilities in assembling and analyzing Sanger sequencing data. In the years 1984–1985, other sequence manipulation suites were developed to run on CP/M, Apple II and Macintosh computers [51, 52]. Some of the developers of those software offered free code copies on demand, thereby exemplifying an upcoming software sharing movement in the programming world.

Bioinformatics and the free software movement

In 1985, Richard Stallman published the GNU Manifesto, which outlined his motivation for creating a free Unix-based operating system called GNU (GNU's Not Unix) [53]. This movement later grew as the Free Software Foundation, which promotes the philosophy that 'the users have the freedom to run, copy, distribute, study, change and improve the software' [54]. The free software philosophy promoted by Stallman was at the core of several initiatives in bioinformatics such as the European Molecular Biology Open Software Suite, whose development began later in 1996 as a free and open source alternative to GCG [55, 56]. In fact, this train of thought was already notable in earlier initiatives that predate the GNU project. Such an example is the Collaborative Computational Project Number 4 (CCP4) for macromolecular X-ray crystallography, which was initiated in 1979 and still commonly used today [57, 58].

Most importantly, it is during this period that the European Molecular Biology Laboratory (EMBL), GenBank and DNA Data Bank of Japan (DDBJ) sequence databases have united (EMBL and GenBank in 1986 and finally DDBJ in 1987) in order, among other things, to standardize data formatting, to define minimal information for reporting nucleotide sequences and to facilitate data sharing between those databases. Today this union still exists and is now represented by the International Nucleotide Sequence Database Collaboration (<http://www.insdc.org/>) [59].

The 1980s were also the moment where bioinformatics became present enough in modern science to have a dedicated journal. Effectively, given the increased availability of computers and the enormous potential of performing computer-assisted analyses in biological fields, a journal specialized in bioinformatics, *Computer Applications in the Biosciences* (CABIOS), was established in 1985. This journal, now named *Bioinformatics*, had the mandate to democratize bioinformatics among biologists:

'CABIOS is a journal for life scientists who wish to understand how computers can assist in their work. The emphasis is on



Illustration 3. An HP-9000 desktop workstation running the Unix-based system HP-UX. Image: Thomas Schanz//CC-BY-SA 3.0.

application and their description in a scientifically rigorous format. Computing should be no more mystical a technique than any other laboratory method; there is clearly scope for a journal providing descriptive reviews and papers that provide the jargon, terms of reference and intellectual background to all aspects of biological computing.’ [60].

The use of computers in biology broadened through the free software movement and the emergence of dedicated scientific journals. However, for large data sets such as whole genomes and gene catalogs, small-scale mainframe computers were used instead of microcomputers. Those systems typically ran on Unix-like operating systems and used different programming languages (e.g. C and FORTRAN) than those typically used on microcomputers (such as BASIC and Pascal). As a result, popular sequence analysis software made for microcomputers were not always compatible with mainframe computers, and vice-versa.

Desktop computers and new programming languages

With the advent of x86 and RISC microprocessors in the early 1980s, a new class of personal computer emerged. Desktop workstations, designed for technical and scientific applications, had dimensions comparable with a microcomputer, but had substantially more hardware performance, as well as a software architecture more similar to mainframe computers. As a matter of fact, desktop workstations typically ran on Unix operating systems and derivatives such as HP-UX and BSD (Illustration 3).

The mid-1980s saw the emergence of several scripting languages that would still remain popular among bioinformaticians today. Those languages abstract significant areas of computing systems and make use of natural language characteristics, thereby simplifying the process of developing a program. Programs written in scripts typically do not require compilation (i.e. they are interpreted when launched), but perform more slowly than equivalent programs compiled from C or Fortran code [61].

Perl (Practical Extraction and Reporting Language) is a high-level, multiparadigm, interpreted scripting language that was created in 1987 by Larry Wall as an addition to the GNU operating system to facilitate parsing and reporting of text data [62]. Its core characteristics made it an ideal language to manipulate biological sequence data, which is well represented in text format. The earliest occurrence of bioinformatics software written in Perl goes back to 1994 (Table 2). Then until the late 2000s, Perl

Table 2. Selected early bioinformatics software written in Perl

Software	Year released	Use	Reference
GeneQuiz	1994 (oldest)	Workbench for protein sequence analysis	[65]
LabBase	1998	Making relational databases of sequence data	[66]
Phred-Phrap-Consed	1998	Genome assembly and finishing	[67]
Swissknife	1999	Parsing of SWISS-PROT data	[68]
MUMmer	1999	Whole genome alignment	[69]

PubMed Key: (perl bioinformatics) AND (“1987”[Date-Publication]:“2000”[Date-Publication]).

was undoubtedly the *lingua franca* of bioinformatics, due to its great flexibility [63]. As Larry Wall stated himself, ‘there’s more than one way to do it’. The development of BioPerl in 1996 (and initial release in 2002) contributed to Perl’s popularity in the bioinformatics field [64]. This Perl programming interface provides modules that facilitate typical but nontrivial tasks, such as (i) accessing sequence data from local and remote databases, (ii) switching between different file formats, (iii) similarity searches, and (iv) annotating sequence data.

However, Perl’s flexibility, coupled with its heavily punctuated syntax, could easily result in low code readability. This makes Perl code maintenance difficult, especially for updating software after several months or years. In parallel, another high-level programming language was to become a major actor in the bioinformatics scene.

Python, just like Perl, is a high-level, multiparadigm programming language that was first implemented by Guido van Rossum in 1989 [70]. Python was especially designed to have a simpler vocabulary and syntax to make code reading and maintenance simpler (at the expense of flexibility), even though both languages can be used for similar applications. However, it was not before the year 2000 that specialized bioinformatics libraries for Python were implemented [71], and it was not until the late 2000s that Python became a major programming language in bioinformatics [72]. In addition to Perl and Python, several non-scripting programming languages originated in the early 1990s and joined the bioinformatics scene later on (Table 3).

The emergence of tools that facilitated DNA analysis, either *in vitro* or *in silico*, permitted increasingly complex endeavors such as the analysis of whole genomes from prokaryotic and eukaryotic organisms since the early 1990s.

1990–2000: Genomics, structural bioinformatics and the information superhighway

Dawn of the genomics era

In 1995, the first complete genome sequencing of a free-living organism (*Haemophilus influenzae*) was sequenced by The Institute for Genomic Research (TIGR) led by geneticist J. Craig Venter [83]. However, the turning point that started the genomic era, as we know it actually, was the publication of the human genome at the beginning of the 21st century [84, 85].

The Human Genome Project was initiated in 1991 by the U.S. National Institutes of Health, and cost \$2.7 billion in taxpayer money (in 1991 dollars) over 13 years [86]. In 1998, Celera Genomics (a biotechnology firm also run by Venter) led a rival,

Table 3. Notable nonscripting and/or statistical programming languages used in bioinformatics

	Fortran ^a	C	R	Java
First appeared	1957	1972	1993	1995
Typical use	Algorithmics, calculations, programming modules for other applications	Optimized command-line tools	Statistical analysis, data visualization	Graphical user interfaces, data visualization, network analysis
Notable fields of application	Biochemistry, Structural Bioinformatics	Various	Metagenomics, Transcriptomics, Systems Biology	Genomics, Proteomics, Systems Biology
Specialized bioinformatics repository?	None	None	Bioconductor, [73], since 2002	BioJava [74], since 2002
Example software or packages	Clustal [32, 33], WHAT IF [75]	MUSCLE [76], PhyloBayes [77]	edgeR [78], phyloseq [79]	Jalview [80], Jemboss [81], Cytoscape [82]

^aEven though the earliest bioinformatics software were written in Fortran, it is seldom used to code standalone programs nowadays. It is rather used to code modules for other programs and programming languages (such as C and R mentioned here).

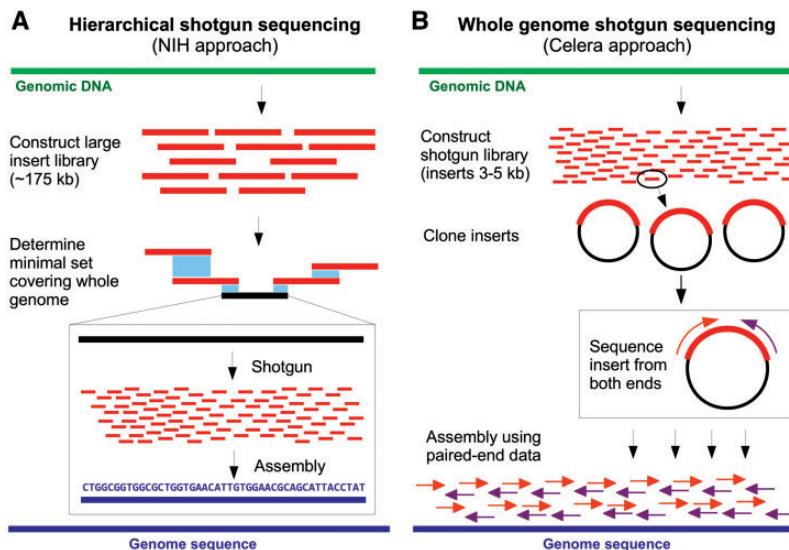


Figure 6. Hierarchical shotgun sequencing versus whole genome shotgun sequencing. Both approaches respectively exemplified the methodological rivalry between the public (NIH, A) and private (Celera, B) efforts to sequence the human genome. Whereas the NIH team believed that whole-genome shotgun sequencing (WGS) was technically unfeasible for gigabase-sized genomes, Venter's Celera team believed that not only this approach was feasible, but that it could also overcome the logistical burden of hierarchical shotgun sequencing, provided that efficient assembly algorithms and sufficient computational power are available. Because of the partial use of NIH data in Celera assemblies, the true feasibility of WGS sequencing for the human genome has been heavily debated by both sides [89, 90].

private effort to sequence and assemble the human genome. The Celera-backed initiative successfully sequenced and assembled the human genome for one-tenth of the National Institutes of Health (NIH)-funded project cost [87]. This 10-fold difference between public and private effort costs resulted from different experimental strategies (Figure 6) as well as the use of NIH data by the Celera-led project [88].

Although the scientific community was experiencing a time of great excitement, whole-genome sequencing required millions of dollars and years to reach completion, even for a bacterial genome. In contrast, sequencing a human genome with 2018 technology would cost \$1000 and take less than a week [91]. This massive cost discrepancy is not so surprising; at this time, even if various library preparation protocols existed, sequencing reads were generated by using Sanger capillary sequencers (Illustration 4). Those had a maximum throughput of 96 reads of 800 bp length per run [92], orders of magnitude less than second-generation sequencers that emerged in the late 2000s. Thus, sequencing the human genome (3.0 Gbp) required a rough minimum of about 40 000 runs to get just one-fold coverage.

In addition to laborious laboratory procedures, specialized software had to be designed to tackle this unprecedented amount of data. Several pioneer Perl-based software were developed in the mid to late 1990s to assemble whole-genome sequencing reads: PHRAP [67], Celera Assembler [93], TIGR Assembler [94], MIRA [95], Euler [96] and many others.

Another important player in the early genomics era was the globalized information network, which made its appearance in the early 1990s. It was through this network of networks that the NIH-funded human genome sequencing project made its data publicly available [97]. Soon, this network would become ubiquitous in the scientific world, especially for data and software sharing.

Bioinformatics online

In the early 1990s, Tim Berners-Lee's work as a researcher at the Conseil Européen pour la Recherche Nucléaire (CERN) initiated the World Wide Web, a global information system made of interlinked documents. Since the mid-1990s, the Web has



Illustration 4. Array of ABI 373 capillary sequencers at the National Human Genome Research Institute (1993). Image: Hank Morgan//SPL v1.0.

revolutionized culture, commerce and technology, and enabled near-instant communication for the first time in the history of mankind.

This technology also led to the creation of many bioinformatics resources accessible throughout the world. For example, the world's first nucleotide sequence database, the EMBL Nucleotide Sequence Data Library (that included several other databases such as SWISS-PROT and REBASE), was made available on the Web in 1993 [98]. It was almost at the same time, in 1992, that the GenBank database became the responsibility of the NCBI (before it was under contract with Los Alamos National Laboratory) [99]. However, GenBank was very different from today and was distributed in print and as a CD-ROM in its first inception.

In addition, the well-known website of the NCBI was made available online in 1994 (including the tool BLAST, which allows to perform pairwise alignments efficiently). Then came the establishment of several major databases still used today: Genomes (1995), PubMed (1997) and Human Genome (1999).

The rise of Web resources also broadened and simplified access to bioinformatics tools, mainly through Web servers with a user-friendly graphical user interface. Indeed, bioinformatics software often (i) require prior knowledge of UNIX-like operating systems, (ii) require the utilization of command lines (for both installation and usage) and (iii) require the installation of several software libraries (dependencies) before being usable, which can be unintuitive even for skilled bioinformaticians.

Fortunately, more developers try to make their tools available to the scientific community through easy-to-use graphical Web servers, allowing to analyze data without having to perform fastidious installation procedures. Web servers are now so present in modern science that the journal *Nucleic Acids Research* publishes a special issue on these tools each year (<https://academic.oup.com/nar>).

The use of Internet was not only restricted to analyzing data but also to share scientific studies through publications, which is the cornerstone of the scientific community. Since the creation of the first scientific journal, *The Philosophical Transactions of the Royal Society*, in 1665 and until recently, the scientists shared their findings through print or oral media.

In the early 1980s, several projects emerged to investigate the possibility, advantages and disadvantages of using the Internet for scientific publications (including submission, revision and reading of articles) [100]. One of the first initiatives, BLEND, a 3-year study that used a cohort of around 50 scientists,

shed light on the possibilities and challenges of such projects [101]. These studies pioneered the use of the Internet for both data set storage and publishing. This usage is exemplified by the implementation of preprint servers such as Cornell University's arXiv (est. 1991) and Cold Spring Harbor's bioRxiv (est. 2013) who perform these tasks simultaneously.

Beyond sequence analysis: structural bioinformatics

The first three-dimensional structure of a protein, that of myoglobin, was determined experimentally in 1958 using X-ray diffraction [102]. However, the first milestones concerning the prediction of a protein structure were laid by Pauling and Corey in 1951 with the publication of two articles that reported the prediction of α -helices and β -sheets [103]. As with other areas in the biological sciences, it is now possible to use computers to perform calculations to predict, with varying degrees of certainty, the secondary and tertiary structure (especially thanks to fold recognition algorithms also called treading) of proteins [104, 105].

Although advances in the field of 3D structure prediction are crucial, it is important to remember that proteins are not static, but rather a dynamic network of atoms. With some breakthroughs in biophysics, force fields have been developed to describe the interactions between atoms, allowing the release of tools to model the molecular dynamics of proteins in the 1990s [106]. Although theoretical methods had been developed and tools were available, it remained very difficult in practice to perform molecular dynamics simulations due to the large computational resources required. For example, in 1998, a microsecond simulation of a 36-amino-acid peptide (villin headpiece subdomain) required months of calculations despite the use of supercomputers with 256 central processing units [107].

Despite the constant increase in the power of modern computers, for many biomolecules, computing resources still remain a problem for making molecular dynamics simulations on a reasonable time scale. Nevertheless, there have been, and continue to be, several innovations, such as the use of graphics processing units (GPUs) through high-performance graphics cards normally used for graphics or video games, that help to make molecular dynamics accessible [108, 109]. Moreover, the use of GPUs also began to spread in other fields of bioinformatics requiring massive computational power, such as the construction of molecular phylogenies [110, 111].

However, the ease brought by the Internet for publishing data, in conjunction with increasing computational power, played a role in the mass production of information we now refer to as 'Big Data'.

2000–2010: High-throughput bioinformatics

Second-generation sequencing

DNA sequencing was democratized with the advent of second-generation sequencing (also called next-generation sequencing or NGS) that started with the '454' pyrosequencing technology [112]. This technology allowed sequencing thousands to millions of DNA molecules in a single machine run, thus raising again the old computational challenge. The gold-standard tool to handle reads from 454 (the high-throughput sequencer) is still today the proprietary tool Newbler, that was maintained by Roche until the phasing out of the 454 in 2016. Now, several

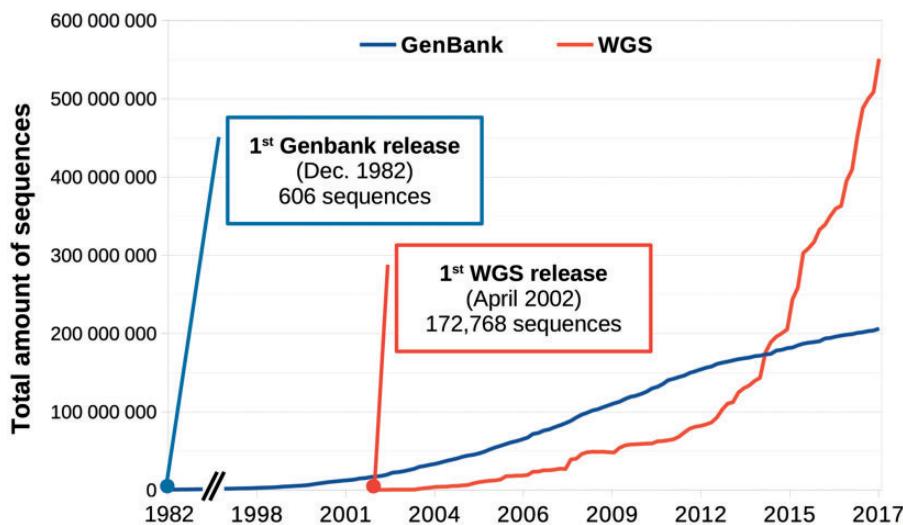


Figure 7. Total amount of sequences on the NCBI GenBank and WGS (Whole Genome Shotgun) databases over time. The number of draft/incomplete genomes has surpassed the amount of complete genome sequences in June 2014, and still continues to grow exponentially. Source: <https://www.ncbi.nlm.nih.gov/genbank/statistics/>.

other companies and technologies are on the market [113] and a multitude of tools is available to deal with the sequences.

In fact, there are now so many tools that it is difficult to choose a specific one. If this trend persists, it will become increasingly difficult for different teams to compare their findings and to replicate results from other research groups. Furthermore, switching to newer and/or different bioinformatics tools requires additional training and testing, thereby making researchers reluctant to abandon software they are familiar with [114]. The mastering of new tools must therefore be justified by an increase in either computing time or results of significantly better quality. This has led to competitions such as the well-known Assemblathon [115], which rates *de novo* assemblers based on performance metrics (assembly speed, contig N50, largest contig length, etc.). However, the increasing amount of available tools is dwarfed by the exponential increase of biological data deposited in public databases.

Biological Big Data

Since 2008, Moore's Law stopped being an accurate predictor of DNA sequencing costs, as they dropped several orders of magnitude after the arrival of massively parallel sequencing technologies (<https://www.genome.gov/sequencingcosts/>). This resulted in an exponential increase of sequences in public databases such as GenBank and WGS (Figure 7) and further preoccupations towards Big Data issues. In fact, the scientific community has now generated data beyond the exabyte (10^{18}) level [116]. Major computational resources are necessary to handle all of this information, obviously to store it, but also to organize it for easy access and use.

New repository infrastructure arose for model organisms such as *Drosophila* [117], *Saccharomyces* [118] and human [119]. Those specialized databases are of great importance, as in addition to providing genomic sequences with annotations (often curated) and metadata, many of them are structuring resources for the scientific community working on these organisms. For example, Dicty Stock Center is a directory of strains and plasmids and is a complementary resource to dictyBase [120], a database of the model organism *Dictyostelium discoideum*. In addition, we have also been witnessing the emergence of general genomic databases such as the Sequence Read Archive [121].

and The European Nucleotide Archive [122], which were implemented to store raw sequencing data for further reproducibility between studies.

Given the large number of genomic sequences and databases that emerge, it is important to have standards to structure these new resources, ensure their sustainability and facilitate their use. With this in mind, the Genomic Standards Consortium was created in 2005 with the mandate to define the minimum information needed for a genomic sequence [123, 124].

High-performance bioinformatics and collaborative computing

The boom in bioinformatics projects, coupled with the exponentially increasing amount of data, has required adaptation from funding bodies. As for the vast majority of scientific studies, bioinformatics projects also require resources. It goes without saying that the most common material for a project in bioinformatics is the computer. Although in some cases and according to the necessary calculations, a simple desktop computer can suffice, some projects in bioinformatics will require infrastructures much more imposing, expensive and requiring special expertise. Several government-sponsored organizations specialized in high-performance computing have emerged, such as:

- Compute Canada (<https://www.computecanada.ca>), which manages the establishment and access of Canadian researchers to computing services;
- New York State's High Performance Computing Program (<https://esd.ny.gov/new-york-state-high-performance-computing-program>);
- The European Technology Platform for High Performance Computing (<http://www.ftp4hpc.eu/>); and
- China's National Center for High-Performance Computing (<http://www.nchc.org.tw/en/>).

The importance of high-performance computing has also led some companies, such as Amazon (<https://aws.amazon.com/health/genomics/>) and Microsoft (<https://enterprise.microsoft.com/en-us/industries/health/genomics/>), to offer services in bioinformatics.

In addition, the rise of community computing has redefined how one can participate in bioinformatics. This is exemplified

by BOINC [125], which is a collaborative platform that allows users to make their computers available for distributed calculations for different projects. Experts can submit computing tasks to BOINC, while nonexperts and/or science enthusiasts can volunteer by allocating their computer resources to jobs submitted to BOINC. Several projects related to life sciences are now available through BOINC, such as for protein–ligand docking, simulations related to malaria and protein folding.

2010–Today: Present and future perspectives

Clearly defining the bioinformatician profession

A recent evolution related to bioinformatics is the emergence of researchers specialized in this field: the bioinformaticians [126]. Even after more than 50 years of bioinformatics, there is still no definite consensus on what is a bioinformatician.

For example, some authors suggested that the term 'bioinformatician' be reserved to those specialized in the field of bioinformatics, including those who develop, maintain and deploy bioinformatics tools [126]. On the other hand, it was also suggested that any user of bioinformatics tools should be granted the status of bioinformatician [127]. Another tentative, albeit more humorous, defined the bioinformatician contrapositively, i.e. how not to be one [128].

What is certain, however, is that there is a significant increase in (i) user-friendly tools, often available through integrative Web servers like Galaxy [129], and (ii) helping communities such as SEQanswers [130] and BioStar [131]. There is also an explosive need for bioinformaticians on the job market in academic, private and governmental sectors [132]. To fill this necessity, universities were urged to adapt the curriculum of their biological sciences programs [133–136].

Now, life scientists not directly involved in a bioinformatics program need to be skilled at basic concepts to understand the subtleties of bioinformatics tools to avoid misuse and erroneous interpretations of the results [136, 137].

The International Society for Computational Biology published guidelines and recommendations of core competencies that a bioinformatician should have in her/his curriculum [133], based on three user categories (bioinformatics user, bioinformatics scientist and bioinformatics engineer). All three user categories contain core competencies such as:

'[using] current techniques, skills, and tools necessary for computational biology practice', '[applying] statistical research methods in the contexts of molecular biology, genomics, medical, and population genetics research' and 'knowledge of general biology, in-depth knowledge of at least one area of biology, and understanding of biological data generation technologies'.

Additional competencies were defined for the remaining two categories, such as:

'[analyzing] a problem and identify and define the computing requirements appropriate to its solution' for the bioinformatics scientist, and '[applying] mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices' for the bioinformatics engineer'.

Is the term 'bioinformatics' now obsolete?

Before attempting to define the bioinformatician profession, perhaps bioinformatics itself requires a proper definition.

Indeed, the use of computers has become ubiquitous in biology, as well as in most natural sciences (physics, chemistry, mathematics, cryptography, etc.), but interestingly, only biology has a specific term to refer to the use of computers in this discipline ('bioinformatics'). Why is that so?

First, biology has historically been perceived as being at the interface of 'hard' and 'soft' sciences [138]. Second, the use of computers in biology required a certain understanding of the structure of macromolecules (namely, nucleic acids and proteins). This led biology to computerize itself later than other 'hard' sciences such as physics and mathematics. This is not so surprising, knowing that the first computers were designed specifically to solve mathematical calculations in the field of physics. For example, one of the first fully electronic computers, the ENIAC (1943–1946), was first used for the development of the hydrogen bomb during World War II.

The combination of these factors might clarify why the connection between biology and computers was not immediately obvious. This could also explain why the use of the term 'Bioinformatics' still remains in common usage. Today, when virtually any research endeavor requires using computers, one may question the relevance of this term in the future. An interesting thought was made to this regard by the bioinformatician C. Titus Brown at the 15th annual Bioinformatics Open Source Conference. He presented a history of current bioinformatics but told from the perspective of a biologist in year 2039 [139]. In Brown's hypothetical future, biology and bioinformatics are so intertwined that there is no need to distinguish one from the other. Both are simply known as biology.

Towards modeling life as a whole: systems biology

The late 20th century witnessed the emergence of computers in biology. Their use, along with continuously improving laboratory technology, has permitted increasingly complex research endeavors. Whereas sequencing a single protein or gene could have been the subject of a doctoral thesis up to the early 1990s, a PhD student may now analyze the collective genome of many microbial communities during his/her graduate studies [140]. Whereas determining the primary structure of a protein was complex back then, one can now identify the whole proteome of a sample [141]. Biology has now embraced a holistic approach, but within distinct macromolecular classes (e.g. genomics, proteomics and glycomics) with little crosstalk between each subdiscipline.

One may anticipate the next leap: instead of independently investigating whole genomes, whole transcriptomes or whole metabolomes, whole living organisms and their environments will be computationally modeled, with all molecular categories taken into account simultaneously. In fact, this feat has already been achieved in a whole cell model of *Mycoplasma genitalium*, in which all its genes, their products and their known metabolic interactions have been reconstructed in silico [142]. Perhaps we will soon witness an in silico model of a whole pluricellular organism. Even though this might seem unfeasible to model millions to trillions of cells, one must keep in mind that we are now achieving research exploits that would have been deemed as computationally or technically impossible even 10 years ago.

Key Points

- The very beginnings of bioinformatics occurred more than 50 years ago, when desktop computers were still a

- hypothesis and DNA could not yet be sequenced.
- In the 1960s, the first *de novo* peptide sequence assembler, the first protein sequence database and the first amino acid substitution model for phylogenetics were developed.
 - Through the 1970s and the 1980s, parallel advances in molecular biology and computer science set the path for increasingly complex endeavors such as analyzing complete genomes.
 - In the 1990–2000s, use of the Internet, coupled with next-generation sequencing, led to an exponential influx of data and a rapid proliferation of bioinformatics tools.
 - Today, bioinformatics faces multiple challenges, such as handling Big Data, ensuring the reproducibility of results and a proper integration into academic curriculums.

Acknowledgements

The authors wish to thank Eric Normandeau and Bachar Cheaib for sharing ideas and helpful insights and criticism. J.G. specifically wishes to thank the staff of the Institut de Biologie Intégrative et des Systèmes (IBIS) for attending his seminars on the history of bioinformatics, presented between 2015 and 2016 to the IBIS community. What began as a series of talks ultimately became the basis for this present historical review.

Funding

J.G. received a Graduate Scholarship from the Natural Sciences and Engineering Research Council (NSERC) and A.T.V. received a Postdoctoral Fellowship from the NSERC. S.J.C. is a research scholar of the Fonds de Recherche du Québec-Santé.

Author contributions

J.G. and A.T.V. performed the documentary research and have written the manuscript. S.J.C. and N.D. helped to draft the manuscript and supervised the work.

References

1. Avery OT, MacLeod CM, McCarty M. Studies on the chemical nature of the substance inducing transformation of pneumococcal types. *J Exp Med* 1944;79:137–58.
2. Griffiths AJ, Miller JH, Suzuki DT, et al. *An Introduction to Genetic Analysis*. Holtzbrinck: W. H. Freeman, 2000, 860.
3. Hershey AD, Chase M. Independent functions of viral protein and nucleic acid in growth of bacteriophage. *J Gen Physiol* 1952;36:39–56.
4. Tamm C, Shapiro HS, Lipshitz R, et al. Distribution density of nucleotides within a deoxyribonucleic acid chain. *J Biol Chem* 1953;203:673–88.
5. Watson JD, Crick FHC. Molecular structure of nucleic acids: a structure for deoxyribose nucleic acid. *Nature* 1953;171:737–8.
6. Nirenberg M, Leder P. RNA codewords and protein synthesis. The effect of trinucleotides upon the binding of sRNA to ribosomes. *Science* 1964;145:1399–407.
7. Sanger F, Nicklen S, Coulson AR. DNA sequencing with chain-terminating inhibitors. *Proc Natl Acad Sci USA* 1977;74:5463–7.
8. Maxam AM, Gilbert W. A new method for sequencing DNA. *Proc Natl Acad Sci USA* 1977;74:560–4.
9. Jaskolski M, Dauter Z, Włodawer A. A brief history of macromolecular crystallography, illustrated by a family tree and its Nobel fruits. *FEBS J* 2014;281:3985–4009.
10. Sanger F, Thompson EOP. The amino-acid sequence in the glycol chain of insulin. I. The identification of lower peptides from partial hydrolysates. *Biochem J* 1953;53:353–66.
11. Sanger F, Thompson EOP. The amino-acid sequence in the glycol chain of insulin. II. The investigation of peptides from enzymic hydrolysates. *Biochem J* 1953;53:366–74.
12. Hagen JB. The origins of bioinformatics. *Nat Rev Genet* 2000;1:231–6.
13. Edman P. A method for the determination of amino acid sequence in peptides. *Arch Biochem* 1949;22:475.
14. Edman P, Begg G. A protein sequenator. *Eur J Biochem* 1967;1:80–91.
15. Moody G. *Digital Code of Life: How Bioinformatics is Revolutionizing Science, Medicine, and Business*. London: Wiley, 2004.
16. Oakley MB, Kimball GE. Punched card calculation of resonance energies. *J Chem Phys* 1949;17:706–17.
17. Ledley RS. Digital electronic computers in biomedical science. *Science* 1959;130:1225–34.
18. November JA. Early biomedical computing and the roots of evidence-based medicine. *IEEE Ann Hist Comput* 2011;33:9–23.
19. Dayhoff MO, Ledley RS. Comprotein: a computer program to aid primary protein structure determination. In: *Proceedings of the December 4–6, 1962, Fall Joint Computer Conference*. New York, NY: ACM, 1962, 262–74.
20. IUPAC-IUB Commission on Biochemical Nomenclature (CBN). A one-letter notation for amino acid sequences*. *Eur J Biochem* 1968;5:151–3.
21. Dayhoff MO; National Biomedical Research Foundation. *Atlas of Protein Sequence and Structure*, Vol. 1. Silver Spring, MD: National Biomedical Research Foundation, 1965.
22. Srinivasan PR. *The Origins of Modern Biochemistry: A Retrospect on Proteins*. New York Academy of Sciences, 1993, 325.
23. Shanon B. The genetic code and human language. *Synthese* 1978;39:401–15.
24. Pinker S, Bloom P. Natural language and natural selection. *Behav Brain Sci* 1990;13:707–27.
25. Pauling L, Zuckerkandl E. Chemical paleogenetics: molecular “restoration studies” of extinct forms of life. *Acta Chem Scand* 1963;17:S9–16.
26. Fitch WM. Distinguishing homologous from analogous proteins. *Syst Zool* 1970;19:99–113.
27. Haber JE, Koshland DE. An evaluation of the relatedness of proteins based on comparison of amino acid sequences. *J Mol Biol* 1970;50:617–39.
28. Needleman SB, Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol* 1970;48:443–53.
29. Murata M, Richardson JS, Sussman JL. Simultaneous comparison of three protein sequences. *Proc Natl Acad Sci USA* 1985;82:3073–7.
30. Wang L, Jiang T. On the complexity of multiple sequence alignment. *J Comput Biol* 1994;1:337–48.
31. Feng DF, Doolittle RF. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J Mol Evol* 1987;25:351–60.

32. Higgins DG, Sharp PM. CLUSTAL: a package for performing multiple sequence alignment on a microcomputer. *Gene* 1988;73:237–44.
33. Sievers F, Higgins DG. Clustal Omega, accurate alignment of very large numbers of sequences. *Methods Mol Biol* 2014; 1079:105–16.
34. Dayhoff MO, Schwartz RM, Orcutt BC. Chapter 22: a model of evolutionary change in proteins. In: *Atlas of Protein Sequence and Structure*. Washington, DC: National Biomedical Research Foundation, 1978.
35. Crick FH. The origin of the genetic code. *J Mol Biol* 1968;38: 367–79.
36. Hert DG, Fredlake CP, Barron AE. Advantages and limitations of next-generation sequencing technologies: a comparison of electrophoresis and non-electrophoresis methods. *Electrophoresis* 2008;29:4618–26.
37. Staden R. A strategy of DNA sequencing employing computer programs. *Nucleic Acids Res* 1979;6:2601–10.
38. Cornish-Bowden A. Nomenclature for incompletely specified bases in nucleic acid sequences: recommendations 1984. *Nucleic Acids Res* 1985;13:3021–30.
39. Haeckel E. *Generelle Morphologie Der Organismen. Allgemeine Grundzüge Der Organischen Formen-Wissenschaft, Mechanisch Begründet Durch Die Von Charles Darwin Reformirte Descendenztheorie*. Berlin: G. Reimer, 1866, 626.
40. Felsenstein J. Evolutionary trees from DNA sequences: a maximum likelihood approach. *J Mol Evol* 1981;17:368–76.
41. Rannala B, Yang Z. Probability distribution of molecular evolutionary trees: a new method of phylogenetic inference. *J Mol Evol* 1996;43:304–11.
42. Nascimento FF, Reis MD, Yang Z. A biologist's guide to Bayesian phylogenetic analysis. *Nat Ecol Evol* 2017;1:1446–54.
43. Berg P, Baltimore D, Brenner S, et al. Summary statement of the Asilomar conference on recombinant DNA molecules. *Proc Natl Acad Sci USA* 1975;72:1981–4.
44. Kleppe K, Ohtsuka E, Kleppe R, et al. Studies on polynucleotides. XCVI. Repair replications of short synthetic DNA's as catalyzed by DNA polymerases. *J Mol Biol* 1971;56:341–61.
45. Mullis KB, Faloona FA. Specific synthesis of DNA in vitro via a polymerase-catalyzed chain reaction. *Methods Enzymol* 1987; 155:335–50.
46. Mullis KB. Nobel Lecture: The Polymerase Chain Reaction. *Nobelprize.org*. 1993. https://www.nobelprize.org/nobel_prizes/chemistry/laureates/1993/mullis-lecture.html (15 February 2018, date last accessed)
47. McKenzie K. A structured approach to microcomputer system design. *Behav Res Methods Instrum* 1976;8:123–8.
48. Roberts E, Yates W. Altair 8800 minicomputer. *Pop Electron* 1975;7:33–8.
49. Kurtz TE. *BASIC. History of Programming Languages I*. New York, NY: ACM, 1981, 515–37.
50. Devereux J, Haeberli P, Smithies O. A comprehensive set of sequence analysis programs for the VAX. *Nucleic Acids Res* 1984;12:387–95.
51. Malthiery B, Bellon B, Giorgi D, et al. Apple II PASCAL programs for molecular biologists. *Nucleic Acids Res* 1984;12: 569–79.
52. Johnsen M. JINN, an integrated software package for molecular geneticists. *Nucleic Acids Res* 1984;12:657–64.
53. Williams S. *Free as in Freedom: Richard Stallman's Crusade for Free Software*. Sebastopol, CA: O'Reilly, 2002, 240.
54. Free Software Foundation. What is free software? The free software definition. *GNU Operating System*, 2018. <http://www.gnu.org/philosophy/free-sw.en.html> (15 February 2018, date last accessed).
55. Rice P, Longden I, Bleasby A. EMBOSS: the European molecular biology open software suite. *Trends Genet* 2000;16:276–7.
56. Rice P, Bleasby A, Ison J, et al. 1.1. History. *EMBOSS User Guide*. <http://emboss.open-bio.org/html/use/ch01s01.html> (22 January 2018, date last accessed).
57. Collaborative Computational Project, Number 4. The CCP4 suite: programs for protein crystallography. *Acta Crystallogr D Biol Crystallogr* 1994;50:760–3.
58. Winn MD, Ballard CC, Cowtan KD, et al. Overview of the CCP4 suite and current developments. *Acta Crystallogr D Biol Crystallogr* 2011;67:235–42.
59. Karsch-Mizrachi I, Takagi T, Cochrane G. The international nucleotide sequence database collaboration. *Nucleic Acids Res* 2018;46:D48–51.
60. Beynon RJ. CABIOS editorial. *Bioinformatics* 1985;1:1.
61. Fourment M, Gillings MR. A comparison of common programming languages used in bioinformatics. *BMC Bioinformatics* 2008;9:82.
62. Sheppard D. Beginner's Introduction to Perl. *Perl.com*. 2000. <https://www.perl.com/pub/2000/10/begperl1.html> (15 February 2018, date last accessed).
63. Sharma V. Chapter 5. Programming languages. In: *Text Book of Bioinformatics*. Meerut, India: Rastogi Publications, 2008.
64. Stajich JE, Block D, Boulez K, et al. The Bioperl Toolkit: Perl modules for the life sciences. *Genome Res* 2002;12:1611–18.
65. Scharf M, Schneider R, Casari G, et al. GeneQuiz: a workbench for sequence analysis. *Proc Int Conf Intell Syst Mol Biol* 1994;2:348–53.
66. Goodman N, Rozen S, Stein LD, et al. The LabBase system for data management in large scale biology research laboratories. *Bioinformatics* 1998;14:562–74.
67. Gordon D, Abajian C, Green P. Consed: a graphical tool for sequence finishing. *Genome Res* 1998;8:195–202.
68. Hermjakob H, Fleischmann W, Apweiler R. Swissknife—'lazy parsing' of SWISS-PROT entries. *Bioinformatics* 1999;15: 771–2.
69. Kurtz S, Phillippe A, Delcher AL, et al. Versatile and open software for comparing large genomes. *Genome Biol* 2004;5: R12.
70. Venners B. The Making of Python: A Conversation with Guido van Rossum, Part I. 2003. <http://www.artima.com/intv/pythonP.html> (26 January 2018, date last accessed).
71. Chapman B, Chang J. Biopython: Python tools for computational biology. *ACM SIGBIO News* 2000;20:15–19.
72. Ekmekci B, McAnany CE, Mura C. An introduction to programming for bioscientists: a Python-based primer. *PLOS Comput Biol* 2016;12:e1004867.
73. Gentleman RC, Carey VJ, Bates DM, et al. Bioconductor: open software development for computational biology and bioinformatics. *Genome Biol* 2004;5:R80.
74. Holland RCG, Down TA, Pocock M, et al. BioJava: an open-source framework for bioinformatics. *Bioinformatics* 2008;24: 2096–7.
75. Vriend G. WHAT IF: a molecular modeling and drug design program. *J Mol Graph* 1990;8:52–56.
76. Edgar RC. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res* 2004;32: 1792–7.
77. Lartillot N, Lepage T, Blanquart S. PhyloBayes 3: a Bayesian software package for phylogenetic reconstruction and molecular dating. *Bioinformatics* 2009;25:2286–8.

78. Robinson MD, McCarthy DJ, Smyth GK. edgeR: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* 2010;26:139–40.
79. McMurdie PJ, Holmes S. phyloseq: an R package for reproducible interactive analysis and graphics of microbiome census data. *PLoS One* 2013;8:e61217.
80. Waterhouse AM, Procter JB, Martin DMA, et al. Jalview Version 2—a multiple sequence alignment editor and analysis workbench. *Bioinformatics* 2009;25:1189–91.
81. Carver T, Bleasby A. The design of Jemboss: a graphical user interface to EMBOSS. *Bioinformatics* 2003;19:1837–43.
82. Shannon P, Markiel A, Ozier O, et al. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res* 2003;13:2498–504.
83. Fleischmann RD, Adams MD, White O, et al. Whole-genome random sequencing and assembly of *Haemophilus influenzae* Rd. *Science* 1995;269:496–512.
84. Venter JC, Adams MD, Myers EW, et al. The sequence of the human genome. *Science* 2001;291:1304–51.
85. Lander ES, Linton LM, Birren B, et al. Initial sequencing and analysis of the human genome. *Nature* 2001;409:860–921.
86. NHGRI. Human Genome Project Completion: Frequently Asked Questions. National Human Genome Research Institute (NHGRI). <https://www.genome.gov/11006943/Human-Genome-Project-Completion-Frequently-Asked-Questions> (25 January 2018, date last accessed).
87. Whitelaw E. The race to unravel the human genome. *EMBO Rep* 2002;3:515.
88. Macilwain C. Energy department revises terms of Venter deal after complaints. *Nature* 1999;397:93. <https://www.nature.com/articles/16312> (21 May 2018, date last accessed).
89. Waterston RH, Lander ES, Sulston JE. On the sequencing of the human genome. *Proc Natl Acad Sci U S A* 2002;99:3712–16.
90. Adams MD, Sutton GG, Smith HO, et al. The independence of our genome assemblies. *Proc Natl Acad Sci USA* 2003;100:3025–6.
91. NHGRI. DNA Sequencing Costs: Data. National Human Genome Research Institute (NHGRI). <https://www.genome.gov/27541954/DNA-Sequencing-Costs-Data> (25 January 2018, date last accessed).
92. Karger BL, Guttman A. DNA sequencing by capillary electrophoresis. *Electrophoresis* 2009;30:S196–202.
93. Myers EW, Sutton GG, Delcher AL, et al. A whole-genome assembly of *Drosophila*. *Science* 2000;287:2196–204.
94. Sutton GG, White O, Adams MD, et al. TIGR assembler: a new tool for assembling large shotgun sequencing projects. *Genome Sci Technol* 1995;1:19–19.
95. Chevreux B, Wetter T, Suhai S. Genome Sequence Assembly Using Trace Signals and Additional Sequence Information. In: *Proceedings of the German Conference on Bioinformatics*. Hannover, Germany: Fachgruppe Bioinformatik, 1999, 1–12.
96. Pevzner PA, Tang H, Waterman MS. An Eulerian path approach to DNA fragment assembly. *Proc Natl Acad Sci USA* 2001;98:9748–53.
97. Stolov SE. Internet: a computer support tool for building the human genome. In: *Proceedings of Electro/International*. Boston, MA: Institute of Electrical and Electronics Engineers (IEEE), 1995, 441–52.
98. Rice CM, Fuchs R, Higgins DG, et al. The EMBL data library. *Nucleic Acids Res* 1993;21:2967–71.
99. Benson D, Lipman DJ, Ostell J. GenBank. *Nucleic Acids Res* 1993;21:2963–5.
100. McKnight C. Electronic journals—past, present ... and future? *Aslib Proc* 1993;45:7–10.
101. Shackel B. The BLEND system programme for the study of some ‘electronic journals’. *Ergonomics* 1982;25:269–84.
102. Kendrew JC, Bodo G, Dintzis HM, et al. A three-dimensional model of the myoglobin molecule obtained by x-ray analysis. *Nature* 1958;181:662–6.
103. Pauling L, Corey RB. Configurations of polypeptide chains with favored orientations around single bonds. *Proc Natl Acad Sci USA* 1951;37:729–40.
104. Dorn M, e Silva MB, Burilo LS, et al. Three-dimensional protein structure prediction: methods and computational strategies. *Comput Biol Chem* 2014;53:251–76.
105. Yang Y, Gao J, Wang J, et al. Sixty-five years of the long march in protein secondary structure prediction: the final stretch? *Brief Bioinform* 2018;19:482–94.
106. Wooley JC, Ye Y. A historical perspective and overview of protein structure prediction. *Computational Methods for Protein Structure Prediction and Modeling*. New York, NY: Springer, 2007, 1–43.
107. Duan Y, Kollman PA. Pathways to a protein folding intermediate observed in a 1-microsecond simulation in aqueous solution. *Science* 1998;282:740–4.
108. Hospital A, Goñi JR, Orozco M, et al. Molecular dynamics simulations: advances and applications. *Adv Appl Bioinform Chem* 2015;8:37–47.
109. Lane TJ, Shukla D, Beauchamp KA, et al. To milliseconds and beyond: challenges in the simulation of protein folding. *Curr Opin Struct Biol* 2013;23:58–65.
110. Ayres DL, Darling A, Zwickl DJ, et al. BEAGLE: an application programming interface and high-performance computing library for statistical phylogenetics. *Syst Biol* 2012;61:170–3.
111. Martins WS, Rangel TF, Lucas DCS, et al. Phylogenetic distance computation using CUDA. In: *Advances in Bioinformatics and Computational Biology*. BSB 2012. Lecture Notes in Computer Science, Vol. 7409. Berlin, Heidelberg: Springer, 2012, 168–78.
112. Margulies M, Egholm M, Altman WE, et al. Genome sequencing in microfabricated high-density picolitre reactors. *Nature* 2005;437:376–80.
113. Goodwin S, McPherson JD, McCombie WR. Coming of age: ten years of next-generation sequencing technologies. *Nat Rev Genet* 2016;17:333–51.
114. Sandve GK, Nekrutenko A, Taylor J, et al. Ten simple rules for reproducible computational research. *PLoS Comput Biol* 2013;9:e1003285.
115. Bradnam KR, Fass JN, Alexandrov A, et al. Assemblathon 2: evaluating *de novo* methods of genome assembly in three vertebrate species. *GigaScience* 2013;2:1–31.
116. Li Y, Chen L. Big biological data: challenges and opportunities. *Genomics Proteomics Bioinformatics* 2014;12:187–9.
117. Gramates LS, Marygold SJ, Santos GD, et al. FlyBase at 25: looking to the future. *Nucleic Acids Res* 2017;45:D663–71.
118. Cherry JM, Hong EL, Amundsen C, et al. *Saccharomyces* genome database: the genomics resource of budding yeast. *Nucleic Acids Res* 2012;40:D700–5.
119. Casper J, Zweig AS, Villarreal C, et al. The UCSC genome browser database: 2018 update. *Nucleic Acids Res* 2018;46:D762–9.
120. Fey P, Dodson RJ, Basu S, et al. One stop shop for everything *Dictyostelium*: DictyBase and the Dicty Stock Center in 2012. In: *Dictyostelium Discoideum Protocols*. Totowa, NJ, Humana Press, 2013, 59–92.
121. Leinonen R, Sugawara H, Shumway M, et al. The sequence read archive. *Nucleic Acids Res* 2011;39:D19–21.
122. Leinonen R, Akhtar R, Birney E, et al. The European nucleotide archive. *Nucleic Acids Res* 2011;39:D28–31.

123. Field D, Sterk P, Kottmann R, et al. Genomic standards consortium projects. *Stand Genomic Sci* 2014;9:599–601.
124. Field D, Garrity G, Gray T, et al. The minimum information about a genome sequence (MIGS) specification. *Nat Biotechnol* 2008;26:541–7.
125. Anderson DP. BOINC: a system for public-resource computing and storage. In: *Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing*. Washington, DC: IEEE Computer Society, 2004, 4–10.
126. Vincent AT, Charette SJ. Who qualifies to be a bioinformatician? *Front Genet* 2015;6:164.
127. Smith DR. Broadening the definition of a bioinformatician. *Front Genet* 2015;6:258.
128. Corpas M, Fatumo S, Schneider R. How not to be a bioinformatician. *Source Code Biol Med* 2012;7:3.
129. Afgan E, Baker D, van den Beek M, et al. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. *Nucleic Acids Res* 2016;44:W3–10.
130. Li JW, Schmieder R, Ward RM, et al. SEQanswers: an open access community for collaboratively decoding genomes. *Bioinformatics* 2012;28:1272–3.
131. Parnell LD, Lindenbaum P, Shameer K, et al. BioStar: an online question & answer resource for the bioinformatics community. *PLoS Comput Biol* 2011;7:e1002216.
133. Welch L, Lewitter F, Schwartz R, et al. Bioinformatics curriculum guidelines: toward a definition of core competencies. *PLoS Comput Biol* 2014;10:e1003496.
132. Levine AG. An explosion of bioinformatics careers. *Science* 2014;344:1303–6.
134. Rubinstein JC. Perspectives on an education in computational biology and medicine. *Yale J Biol Med* 2012;85:331–7.
135. Koch I, Fuellen G. A review of bioinformatics education in Germany. *Brief Bioinform* 2008;9:232–42.
136. Vincent AT, Bourbonnais Y, Brouard JS, et al. Implementing a web-based introductory bioinformatics course for non-bioinformaticians that incorporates practical exercises. *Biochem Mol Biol Educ* 2018;46:31–8.
137. Pevzner P, Shamir R. Computing has changed biology–biology education must catch up. *Science* 2009;325:541–2.
138. Smith LD, Best LA, Stubbs DA, et al. Scientific graphs and the hierarchy of the sciences: a Latourian survey of inscription practices. *Soc Stud Sci* 2000;30:73–94.
139. Brown TC. A history of bioinformatics (in the Year 2039). In: 15th Annual Bioinformatics Open Source Conference. Boston, MA: Open Bioinformatics Foundation, 2014, Video, 57 min.
140. Deane-Coe KK, Sarvary MA, Owens TG. Student performance along axes of scenario novelty and complexity in introductory biology: lessons from a unique factorial approach to assessment. *CBE Life Sci Educ* 2017;16:ar3, 1–7.
141. Shin J, Lee W, Lee W. Structural proteomics by NMR spectroscopy. *Expert Rev Proteomics* 2008;5:589–601.
142. Karr JR, Sanghvi JC, Macklin DN, et al. A whole-cell computational model predicts phenotype from genotype. *Cell* 2012;150:389–401.