CHAPTER SEVEN

DNA computing-based Big Data storage

Deepak Sharma and Manojkumar Ramteke
Department of Chemical Engineering, Indian Institute of Technology Delhi, New Delhi, India

Contents

Abstract

In the current digital age, the rate of digital data generation is growing exponentially. Though the capacity of conventional storage devices is continuously increasing, it is far from matching the current exponential growth rate of digital data generation. Further, there is an urgent need for a high-density and high-capacity medium to store the information for a prolonged period. Deoxyribonucleic acid (DNA) seems to be a favorable alternative for storing such exponentially growing digital information for a prolonged period with high density and capacity as it keeps the information at a molecular level using nucleotides. DNA stores genetic information of all living things and the information is transferred from one generation to another accurately due to its precise Watson–Crick base pairing. Researchers have successfully used DNA for storing digital data, which opened the possibility of storing Big Data using DNA-based systems. In this chapter, different conventional tools and challenges associated with Big Data storage are reviewed. Further, the various encoding and encryption methods used for DNA-based data storage are critically analyzed. In addition, the challenges for DNA-based Big Data storage are reviewed, and the capabilities of different approaches to overcome these shortcomings are discussed.

Advances in Computers, Volume 129
ISSN 0065-2458
https://doi.org/10.1016/bs.adcom.2022.08.005

Abbreviations

A	adenine
bp	base pair
C	cytosine
DCN	dynamic circuit network
DNA	deoxyribonucleic acid
G	guanine
GIS	geographic information systems
GPS	global positioning system
HDFS	Hadoop distributed file system
HPCC	high-performance computing cluster
IoT	Internet of Things
MB	megabyte
NoSQL	nonstructured query language
PCR	polymerase chain reaction
RNA	ribonucleic acid
SAT	satisfiability
SQL	structured query language
T	thymine

1. Introduction

In the current digital age, the amount of data is growing at an exponential rate [1–3]. This is referred to as Big Data. The analysis of Big Data is becoming increasingly important primarily due to its widespread applications. It is used for the Internet of Things (IoT) [4–6], for global positioning system (GPS) tracking [7], wearable sensors [8], and smart grids [5]. In healthcare, a quite significant amount of data is collected daily from the laboratory, monitoring sensors, and patients' medical history [8–10]. The healthcare-generated data is used for making personalized medicine [3]. This gives appropriate treatment at a reduced cost. Big Data analysis can be used for identifying the affected area from some diseases [8–10]. This can help in timely planning for diagnosis, and vaccination. In production and manufacturing, zero downtime and transparency can be gained by utilizing Big Data analysis. The government uses Big Data to increase the use of limited resources and services. Data collected from various sensors planned at different public infrastructure locations, such as water supplies, power supplies, etc., is used to identify the utilization rate in different areas [11–13]. Big Data analysis not only helps in proper distribution but also helps in improving government facilities. The data extracted from different government departments play an important role in e-governance in their respective areas [11–13]. Various areas such as cyber cell, cybersecurity, tax, traffic supervision, weather collect the Big Data and help improve the facilities. Another important area of the Big Data application is transportation and logistics [5–7]. Transporters equipped with radio frequency identification generate the data used for planning and managing the routes and maintaining the staff's track record [9–13].

Today, humans live in a world filled with information all around. The information transfer from one generation to another has helped humanity evolve to a current level of development. Documentation of essential data allows the next generation to use that knowledge to make it better. For instance, up to 50,000 years ago, the Paleolithic inhabitants' circumstances and situations were understood by the cave paintings present [14]. A naturally preserved man died in the Neolithic age (7000–4000 years ago) discovered in 1991 in Austrian–Italian Alps named Ötzi [15]. Ötzi opened many ways to understand the anatomic and pathogenic discoveries stored in genetic material over 5000 years. Likewise, today people are generating a considerable amount of digital data that needs to be stored and archived, if possible, for all time [16]. Even though a wide range of data storage technologies is present, current data storage methods will turn outdated due to the limited prolonged existence of the hardware used for storing digital data and the high associated cost [17,18].

The rate of digital data production is increasing the demand for data storage [17] exponentially. As of February 20, 2021, the World Wide Web indexed webs comprised of at least 55 Billion web pages [19]. Although only a small fraction of the generated digital data is needed to be reserved, fulfilling the growth in archiving information is becoming difficult using conventional storage technologies [20,21]. Moreover, the attributes such as data density, data access speed, data retention time, resting, and accessing energy cost of data are becoming extremely crucial [20,22]. This inspired the researchers to work on new data storage technologies that can preserve the data efficiently over an extended period of time.

Researchers are now looking at deoxyribonucleic acid (DNA) as a material for data storage to meet the increasing demand for digital data storage. Neiman [23] proposed the idea of storing digital data on DNA. The reason for choosing DNA for storing the data is primarily because of its proven ability to store biological information efficiently. The genetic information of organisms is transferred from one generation to another through DNA or

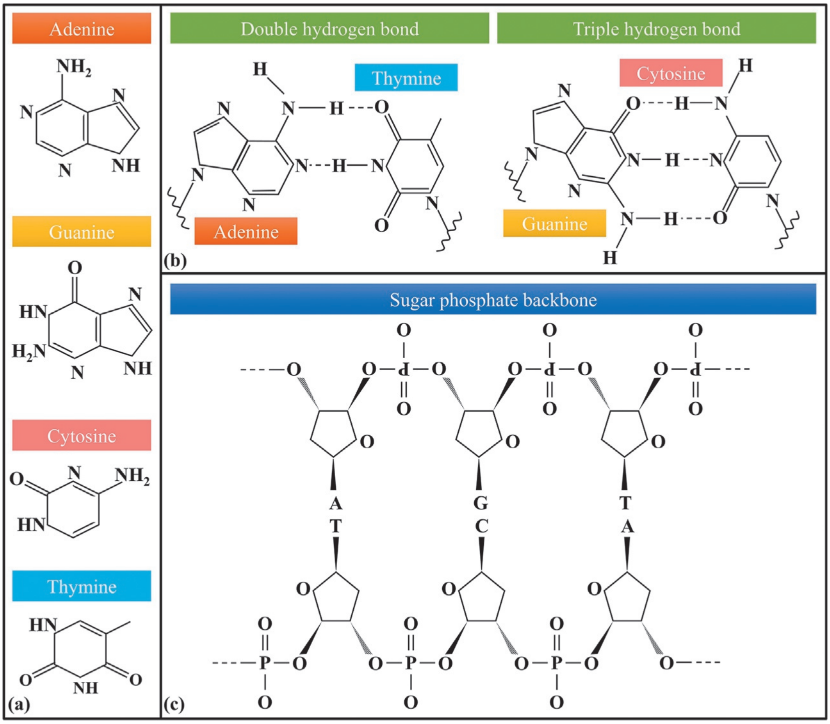

Fig. 1 (a) Nucleotide bases Adenine (A), Guanine (G), Cytosine (C), and Thymine (T), (b) base pairing of A & T, and G & C of two opposite strands by hydrogen bonds (shown by dotted lines), (c) hybridization of two antiparallel and complementary DNA strands.

ribonucleic acid (RNA). As shown in Fig. 1, Adenine (A), Guanine (G), Cytosine (C), and Thymine (T) are the four basic building blocks of the DNA which are called nucleotides [24–26]. Like the digital computers that store the information in binaries (0 and 1), DNA stores the biological information in the quaternary system of four nucleotides (A, T, G, and C). Further, specific DNA sequences used for storing the data can be easily synthesized chemically at a reasonably low cost up to a few hundred base pairs in nanomolar quantities. Moreover, the durability of the DNA is very high as compared to the other biomolecule. The absence of the 2′-hydroxyl group gives a more stable phosphodiester bond in DNA. Under physiological conditions, the half-life of hydrolysis of phosphodiester bond is around 30,000,000 years [27]. The above advantages make DNA an attractive choice for data storage.

The hybridization of individual nucleotides (i.e., A joins only with T and G joins only with C) of two single-stranded DNA is selective (see Fig. 1). This property of selective hybridization serves in identifying the molecules accurately and is referred to as "Watson–Crick complementarity base pairing." The precise Watson–Crick base pairing in DNA is the central property for different types of manipulations such as addition, amplification, cutting, and separation of the DNA. The process of joining two DNA strands is ligation. Two DNA stands join with each other by forming the phosphodiester bond using the ligase enzyme. The method of amplifying the DNA is polymerase chain reaction (PCR), where the formation of multiple copies of the DNA takes place in the presence of the enzyme Taq polymerase. Cutting of the DNA at the precise location is performed using the restriction enzyme digestion. Further, the desired size of DNA is separated using gel electrophoresis, whereas the affinity separation separates the DNA molecules if a specific sequence is present in the DNA molecule. Also, there are numerous high-throughput sequencing methods for sequencing the DNA. In these methods, the DNA strand of interest works as a template for the DNA polymerase reaction. It generates a complementary sequence of the strand comprising fluorescent nucleotides, which emit different colors to read the complementary sequence optically.

The data storage density of DNA is up to 10^{18} bytes per mm^3. This density is six times higher than today's densest media present [21]. The data storage using DNA also enables the protection of the data in molecules at a low cost for an extended period [28]. It has been verified and tested by the time that it is possible to read DNA sequences from thousands of years old fossils [28]. Further, DNA can remains for millennia compared to decades, a usual storage time for media archives when maintained under specific light humidity and temperature [12,18].

Rapid progress in synthesizing and reading DNA establishes DNA as viable storage material, particularly for archival storage, which incurs a high maintenance cost using conventional technology [2,29]. It is useful when duplication of the data is required as the DNA-based storage devices can make millions of copies of the DNA at a low cost and time using PCR [21,30–32], unlike the proportionate time required in conventional technologies. With these advantages, DNA can provide a solution for Big Data storage.

Researchers have successfully used DNA to store digital data up to a megabyte scale [21,33–35]. Further, the encapsulation in silica may help in protecting the data for millennia-long [28]. Also, DNA is a highly dense material that can work as a data archive [17,30,36]. Recently, Erlich and Zielinski [37] practically demonstrated high-density data storage. These methods are then scaled up to store ~200 megabytes (MB) data on DNA by Organick et al. [38]. Microsoft and the University of Washington [32] developed and demonstrated a fully automated machine that can encode digital data on DNA and can decode it back with high accuracy. The researchers [39–41] have generated DNA computing-based passwords for confidential multimedia files or messages to enhance security. This model's advantage is that it provides useful and robust access control to the user to minimize the data access time with high confidence in the cloud-based environment [40–42]. The model also helps in data management and improvement in the works related to healthcare like telemedicine and virtual medication [8]. Several of the above studies and their limitations are listed in Table 1. The practical demonstration of data storage on DNA in the above studies opened the possibility of storing Big Data on DNA. Though several conventional tools and technologies are available for analyzing and storing Big Data, the use of molecular computing methods such as DNA computing and DNA-based storage has the potential to address critical issues such as the exponential growth of Data and its prolonged storage. This motivated the authors to review the existing technologies for Big Data storage, the challenges associated with conventional Big Data storage methodologies, and possible solutions using DNA-based storage methods and their associated challenges.

The significant contributions of this chapter are listed below.

1. It provides a review on DNA-based computing and data storage
2. It offers a review of challenges and tools used for Big Data storage and analytics
3. It gives a detailed experimental procedure of DNA-based data storage
4. It provides a critical review on different DNA-based storage methods, their advantages, and limitations for Big Data storage

This chapter focuses on the analysis of various encoding and encryption methods used for DNA-based data storage. Section 2 reviews the background studies on DNA-based computing, types, and characteristics of Big Data, and technologies and tools used to manage Big Data and their associated challenges. The encoding, decoding methods, and experimental procedure to store the digital data on DNA are presented in Section 3. Challenges for DNA-based Big Data storage are discussed in Section 4, followed by conclusions and future work in Section 5.

Table 1 A literature review of DNA-based data storage.

Study	Total data (MB)	DNA synthesis by	DNA sequencing by	Coverage	Reassembly	Length of DNA strands (base pair)	Bits per base with primer	Bits per base without primer	Random access	Limitations
Church et al. [36]	0.65	Deposition	Illumina	3000 ×	Index	115	0.6	0.83	No	PCR amplification is required multiple times for sequencing to reduce reading error
Goldman et al. [35]	0.63	Deposition	Illumina	51 ×	Overlap	117	0.19	0.29	No	PCR amplification is required multiple times, and the cost of synthesis and sequencing is high
Grass et al. [28]	0.08	Electrochemistry	Illumina	372 ×	Index	158	0.86	1.16	No	DNA synthesis and the sequencing error rates are estimated to be ~1% per nucleotide
Bornholt et al. [30]	0.15	Electrochemistry	Illumina	40 ×	Index	117	0.57	0.85	Yes	DNA synthesis and sequencing are restricting the feasibility of using it in the existing state
Erlich and Zielinski [37]	2	Deposition	Illumina	10.5 ×	Luby seed	152	1.18	1.55	No	DNA sequences created by synthesis and amplification are unequal and uneven. Indexing limits the size of the DNA

Continued

Table 1 A literature review of DNA-based data storage.—cont'd

Study	Total data (MB)	DNA synthesis by	DNA sequencing by	Coverage	Reassembly	Length of DNA strands (base pair)	Bits per base with primer	Bits per base without primer	Random access	Limitations
Blawat et al. [22]	22	Deposition	Illumina	160 ×	Index	230	0.89	1.08	No	The length of the DNA is limited
Organick et al. [38]	200	Deposition	Illumina	5 ×	Index	150–200	0.81	1.1	Yes	Cost is high, and throughput is low
Anavy et al. [43]	8.5	Deposition	Illumina	164 ×	Index	194	1.94	2.64	No	Prone to errors in DNA synthesis and sequencing
Choi et al. [44]	0.000854	Column	Illumina	250 ×	Index	85	1.78	3.37	No	Prone to errors in DNA synthesis and sequencing
Yazdi et al. [45]	0.003	Column	Nanopore	200 ×	Index	880–1000	1.71	1.74	Yes	Prone to errors in DNA sequencing
Organick et al. [38]	0.033	Deposition	Nanopore	36 ×	Index	150	0.81	1.1	Yes	Cost is high, and throughput is low
Lee et al. [46]	1.80E-05	Column (Enzymatic)	Nanopore	175 ×	NA	150–200	1.57	1.57	No	It acquires a sixfold loss in the volumetric density of data

2. Background studies

The DNA storage workflow involves encoding digital information into DNA sequences. For this purpose, suitable DNA sequences are synthesized. These DNA sequences are substantially acclimatized into a library for long-term data storage, retrieving, and random access. The DNA sequencer reads these DNA sequences. The reading (output in fasta format) of the DNA sequencer is in the form of quaternary data ("A," "T," "G," and "C") which is converted back to digital binary data ("0" and "1"). To illustrate the workflow for DNA-based Big Data storage, the fundamentals of DNA computing and Big Data analytics are described next.

2.1 DNA-based computing

Adleman [47] is the first researcher to start using DNA for computing. He solved the Hamiltonian path problem for a supergraph with 7 vertices and 14 edges. The objective was to obtain a path in a supergraph such that all vertices are visited, and each one is visited only once. For this purpose, he represented the vertices and the edges of a given supergraph using single-stranded DNA sequences. The sequences of edges are selected as complementary to half of the predecessor vertex and the remaining half to the successor vertex. These single-stranded DNA sequences ligate selectively as only specific complementary sequences are used in edges to form double-stranded DNA in such a way that these represent only the possible paths present in the given supergraph. From this initial pool of double-stranded DNA, the correct length DNA molecules that satisfy the Hamiltonian path's length are separated using the gel electrophoresis separation. These separated DNA molecules are amplified using polymerase chain reaction (PCR) for better accuracy. Though these separated DNA molecules may have a correct length, some of the vertices may be missing, and some may be repeated multiple times. Therefore, the sequences are further subjected to an affinity separation process that extracts only those paths representing the DNA sequences corresponding to each vertex of the supergraph only once. Thus, after ligation, gel electrophoresis, PCR, and affinity separation, if a DNA molecule is present in the final solution, it represents the Hamiltonian path of a given supergraph. Otherwise, it proves that no Hamiltonian path is possible for the given supergraph.

After the classic study by Adleman [47], the development of several exciting adaptations occurred for solving the computationally intractable problems using DNA. Lipton [48] developed a DNA computer to solve the satisfiability (SAT) problem. Subsequently, Smith et al. [49] and Liu et al. [50] created a new surface-based DNA computing model to solve the SAT problem. The researchers [51–53] have developed the modified approaches for solving the other nondeterministic polynomial-time complete (NP-complete) problems such as 3-SAT, chess, and maximal clique. Sakamoto et al. [54] developed an interesting method to solve the SAT problem using DNA hairpin formation. Sharma and Ramteke [55] solved the practical example of polymer grade scheduling using a DNA computer. Recently, Chao et al. [56] solved a maze problem using a DNA navigator. Most of these studies are illustrated on small-scale problems. To improve the scalability, Sharma and Ramteke [57] developed a DNA computer based on circular DNA structure formation to solve a Hamiltonian cycle problem of the 18-vertex supergraph. In this model, the single-stranded DNA represents the vertices and the supergraph edges. The formation of a circular double-stranded DNA represents a possible path after ligation and hybridization. The restriction enzyme corresponding to each vertex cuts the circular DNA and makes it a linear double-stranded DNA. The obtained linear DNA is then separated using gel electrophoresis. The correct circular DNA cuts only at one location, and the incorrect DNA cuts at multiple locations or remain circular. The iterative circularization (using ligation), cutting by respective restriction enzyme for each vertex, and separation by gel electrophoresis for correct length yields the right solution if it exists after N such steps for N-vertex problem (see Fig. 2). Most of the above studies are reviewed extensively by Sharma and Ramteke [58].

2.2 Big Data analytics

2.2.1 Types of Big Data

Big Data analysis is performed on various data collected from multiple sources such as network, event, time series, natural language, and real-time data. The collected data is in the structured or unstructured form, which is continuously growing. The distribution of this data for analysis turns out to be a crucial step in today's era. The data is noisy and unevenly distributed, which requires some preprocessing steps for analysis; otherwise, data can be misinterpreted. In this section, we are discussing the different data types.

(i) Structured data

Data stored in a proper format that can be readily utilized by a person or computer program is referred to as structured data. A typical example is numerical data stored in the form of a table with labeled rows and columns.

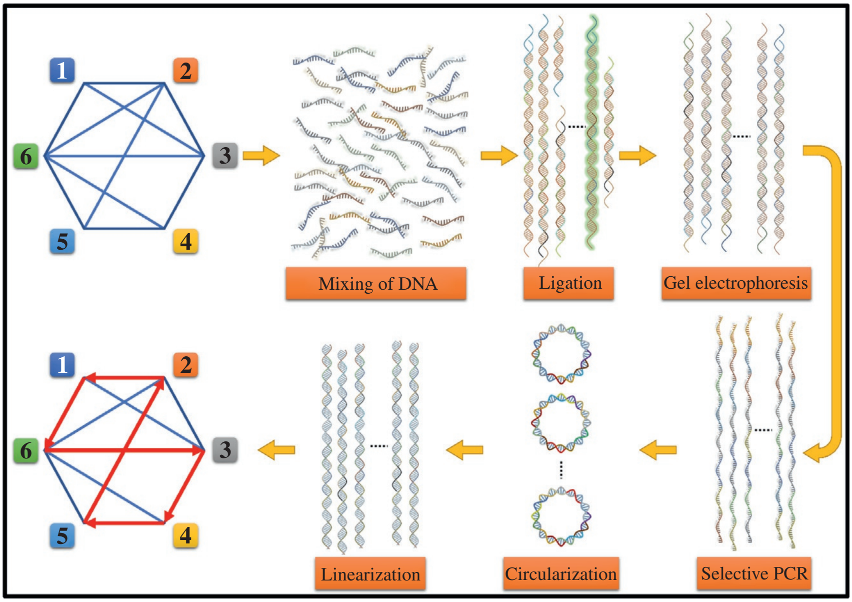

Fig. 2 Circular structure-based DNA computing for solving the Hamiltonian cycle problem.

Database management system (DBMS) facilitates the access of structured data. Only 10% of the total data fit into the structured data generated from various government departments, real estate, different organizations, and transactions.

(ii) Unstructured data

Any data which is not arranged in a predefined form, or a predefined data model is referred to as unstructured data. About 90% of the total data belongs to unstructured data. Unstructured data is in the form of text, audio, video, or image. Growth in social media resulted in the explosion of unstructured data in the past decades. Conventional storage devices and techniques are not sufficient for handling unstructured data. A suitable database [for example, nonstructured query language (NoSQL)] is required for storing and handling this type of data.

Both structured and unstructured data can be further classified based on its source as follows.

(a) Real-time data

Various sources such as images, audio, and video generate real-time data by streaming live media data. Due to the live streaming, storing, and processing, real-time media turn out to be a big challenge. YouTube, flicker, and videoconference are some of the examples of real-time media data generating resources.

(b) Geographic data

Geographic information systems (GISs) generate data such as an address, building, route, workplace, etc. fits into the geographic data. This information is collected from the GIS and sensors installed. The collected raw data is processed to provide valuable information. Environmental conditions are also analyzed by using geographic data.

(c) Natural language

Verbal conversation generates data. This data is known as natural language data. The major sources of natural language data generation are the Internet of Things, speech capturing devices, landlines, and mobile phones.

(d) Network data

Network data is generated from large networks such as Facebook, Twitter, and YouTube. Along with this biological network, information networks are also a few major sources of network data generation. In network data, the connection among the nodes can be established in two forms, first, one-to-one, second, one-to-many. A major challenge in network data is maintaining the connection among the nodes and network architecture.

2.2.2 Characteristics of Big Data

Big Data refers to exponentially growing data sets containing the heterogeneous formats generated from internet sources, social media, social networks, sharing data, sensors, and clickstreams [7,59]. For Big Data applications, conventional business intelligence tools are not efficient. Robust technology and advanced algorithms are required for processing Big Data due to their complex nature. Due to this, researchers and Big Data scientists have defined Big Data using the 3V characteristics [5,12,59]. The next section discusses these 3V characteristics of Big Data.

The first characteristic is *volume*. Today millions of devices (social networks, smartphones, sensors, bar codes, etc.) use the large number of applications that are generating a massive volume of digital data continuously [5]. McAfee et al. [60] forecasted about the generation of digital data on each day in 2012 is approximately 2.5 EB. International Data Corporation published a study estimating the production, replication, and usage of 4.4 zettabytes (ZB) of digital data in 2013. The volume of the data is doubling every 2 years. In 2015, it was 8 ZB [61], and was estimated that it would reach 40 ZB in 2020.

The second characteristic is *velocity*. Digital data is generated at a high rate and needs to be processed instantly to extricate useful information and appropriate insights. YouTube is one of the good examples of the fast speed of Big Data. Another example is Walmart which generates more than 2.5 petabytes (PB) of digital data every hour from its customer's transactions [5].

The third characteristic is *variety*. Audio, videos, documents, etc., are the different formats of the digital data generated by the various resources. This data may be public or private, organized or unorganized, local or distant, shared or confidential global, complete or incomplete, shared or secret, etc. [5,12].

For a better definition of Big Data, some more Vs with extended characteristics are added in the above list by Emani et al. [4] and Gandomi and Haider [62]. These include *vision* (purpose of data), *verification* (confirmation of the data to some specification), *validation* (fulfillment of the purpose of data), *value* (extracting the information from the data for other sectors), and *complexity* (difficulty in organizing and analyzing the data due to evolving relationships).

2.2.3 Technologies and tools to manage Big Data

Big Data management is becoming increasingly important. However, the raw data generated from various sources are messy, noisy, and unduplicable. This type of data management issue is resolved by using appropriate tools and technologies. There are three layers in the usual Big Data project: infrastructure, computing, and application layer. The infrastructure layer operates the interaction between the systems, data storage, and devices in the network. It also organizes the data retrieval requests received from the other layers in the architecture. The computing layer offers an abstraction on the infrastructure layer and the access, organization, and retrieval of data. The data is indexed in this layer and organized over the allocated storage devices. The data is managed over the available repositories after dividing them into blocks. The application layer consists of tools, techniques, and logic for the development of area-specific analytics. The prominent tools and techniques used are presented next.

(i) Apache Hadoop

Apache Hadoop is one of the most famous and used Big Data tools which stores petabytes of structured and unstructured data [63,64]. MapReduce model-based structure of Hadoop works in a distributed environment. Especially, Hadoop is an appropriate choice for the storage of huge amounts of datasets. Data storage and computation are the two fundamental tasks in

Hadoop. Hadoop distributed file system (HDFS) has data storage accountability, and the MapReduce framework usually performs data computations.

(ii) Apache Spark

Apache Spark is an open-source, fast cluster computing framework written in Scala [63–65]. It offers an interface for whole cluster programming, inherent data parallelism, high scalability, and fault tolerance. Though MapReduce-based platforms are suitable for managing large-scale applications, iterative tasks across parallel processes cannot be handled efficiently if an acyclic flow model is applied. Apache Spark enables the implementation of iterative as well as interactive data analysis. It is important to locate the HDFS in query processing using a structured query language (SQL) in Big Data analytics.

(iii) Hadoop's SQL Tools

Hadoop's SQL tools are effective for managing interactive and complex inquiries on big datasets [64]. Therefore, in recent years multiple SQL tools querying on Hadoop tools have been developed. Stream processing systems are service offerings that provide important information to the consumers and empower businesses to make suitable and quick judgments. Tools of stream computing process the live streaming and produce low potential outputs. The applications are denoted in directed acyclic graphs with operators as vertices and flow channels between the operators as edges.

(iv) Apache Storm

Apache Storm is developed for better benchmark processing [65]. It empowers developers to create real-time distributed processing systems. It is also called Hadoop for real-time data. It is highly scalable, user-friendly and can process the unbounded streams of data extremely fast with low latency and guaranteed data processing.

(v) High-Performance Computing Cluster System

High-Performance Computing Cluster (HPCC) System is an open-source computing platform [66]. HPCC helps in managing Big Data workflow. This system is a single platform with a single architecture. Data simulation is performed on the HPCC system in a single programming language.

(vi) Qubole

Qubole is a cloud-based Big Data platform developer that permits customers to formulate, incorporate, and analyze Big Data [67]. Customers of Qubole can connect to Microsoft Excel, Micro Strategy, Tableau, and Qlik through Open Database Connectivity (ODBC) tools. It also offers massively parallel processing.

(vii) Cassandra

Cassandra is a distributed, and open-source database storage system used to manage a huge amount of structured data [68,69]. The structured data is distributed among many commodity servers. The major advantage of Cassandra is that it provides high service with no single point of failure. Cassandra intends to operate over an arrangement of hundreds of nodes possibly distributed around different data centers.

The key features of the above technologies and the tools are listed in Table 2.

Table 2 Technologies and tools to manage Big Data [3,5,7,12].

Sr. no.	Names of the tool	Key features
1.	Apache Hadoop	• Open-source, free license tool • A high-end server is not required • Manages Volume, Variety, Velocity, and Value characteristics • Cost-effective • Scalable with data size • Flexible for different types of data • Involves faster data processing • High fault tolerance resulting nearly no data loss • It offers a robust ecosystem for data • Hadoop Distributed File System (HDFS) can store up to tens of Petabytes of data
2.	Apache Spark	• Open-source tool • Reusable for batch processing • In-memory processing • Cost-efficient • Dynamic in nature • A faster and real-time stream processing tool • High fault tolerance resulting nearly no data loss
3.	Hadoop's SQL tool	• Open-source tool • Compatible with various data sources such as Hive, Avro, Parquet, Optimized Row Columnar (ORC), Java Script Object Notation (JSON), Java Database Connectivity (JDBC)

Continued

Table 2 Technologies and tools to manage Big Data [3,5,7,12].—cont'd

Sr. no.	Names of the tool	Key features
4.	Apache Strom	• Better benchmark processing • Can process up to one million 100-byte messages per second per node • Performs massive parallel calculations • In case a node dies, it automatically restarts on an alternative node • Each unit of data processed at least once • One of the easiest tools for Big Data analysis
5.	High-Performance Computing Cluster System	• Open-source tool • Used for rigorous data computing • It is built on one platform, one architecture, and one programming language • Enhances Scalability, Performance, and Accessibility of the data • Automatically performs the optimization of parallel processing
6.	Qubole	• Cloud-based open-source tool • Autonomous Big Data management platform • Optimizes performance inherently
7.	Cassandra	• NoSQL, distributed, and open-source database • Used to handle large data within multiple servers • Assures high availability with no failure • Replicates data in multiple locations or clouds to ensure the fault tolerance

2.2.4 Challenges of Big Data

The efficient utilization of Big Data may lead to a new trend of valuable progress [3]. Businesses are now shaping to take advantage of Big Data analysis in an extremely competitive environment. New employers focus on attracting employees with proficiencies in dealing with Big Data. It improves operating efficiency, consumer services, and obtaining the updated planned direction, new consumers, and new products and marketplaces. It seems to be a promising approach for dealing with various appealing opportunities. However, the users face multiple challenges as the extent of information exceeds the channeling capabilities. The data is increasing exponentially, whereas improvement in the data processing speed is bounded. There are limited tools available that mainly focus on Big Data analysis. The real-life problem of data storage, sharing, searching, and real-time analysis is challenging for state-of-the-art methods and technologies such as Hadoop.

The challenges associated with Big Data analysis are described below.

(i) Data understanding

Understanding the data is one of the major challenges in Big Data storage. Most of the users do not understand the details of the data. Handling Big Data leads to some open questions such as what is the size of the data? What is the source of the data? How much important the present data is? What type of hardware is required for handling the data? Therefore, an organization needs to find the answers to the above questions for a better understanding of the data.

(ii) Big Data tool selection

After a clear understanding of Big Data, the user should choose an appropriate tool for handling the data. For example, which tool can perform better for analysis of given data (i.e., Hadoop MapReduce, or Apache Spark)? Which tool is better for storage of given data [i.e., Cassandra or Hadoop database (HBase)]? A bad decision on selecting the tools results in loss of data, time, effort, and money.

(iii) Data acquisition

Data acquisition is the process of collecting, sorting, and cleaning the data before transferring it to the storage device. The data acquisition is governed by characteristics such as volume, velocity, variety, and value. Data is acquired from various sources such as media, sensors, social networks, etc., at a considerable rate in structured and unstructured forms (e.g., video, text, pictures, etc.). However, the analysis often requires the data in a specific format. Although the available data can be converted to a particular format, there is always a risk of losing some data in the conversion process.

(iv) Data preprocessing

Before analyzing the data, it should be constructed correctly and represented efficiently. Preprocessing is essential to enhance data quality. It results in improved analysis. Data generated from different sources is extensive and usually, it is unreliable, incomplete, noisy, and inconsistent. The noise can be removed, and inconsistencies can be corrected by applying various data preprocessing methods. Data preprocessing methods include data integration, cleaning, reduction, and transformation.

(v) Data storage

The selection of an appropriate Big Data storage is a critical decision. Attaining consistency in data accessing is also an important factor for selecting data storage. Data storage architecture should give information about the storage facility with the available and reliable storage space, dynamically. Also, access to the data should be provided while analyzing the Big Data. Currently, data storage tools are categorized as direct-attached storage (DAS) and network storage. Network storage can be further classified as storage area network (SAN) and network-attached storage (NAS).

(vi) Data sharing and data transfer

Processing and analysis of Big Data require reliable data sharing and transfer. Intra-dynamic circuit network (DCN) transmission and Inter-DCN transmissions are two types used for data transmission. Intra-DCN transmits the data within the data center, whereas inter-DCN transmits the data from the source to the data center. The data grows continuously, but the capacity of wired and wireless hardware does not increase accordingly. The limited growth of hardware ultimately restricts the complicated data flow across the network and its storing and sharing. This problem is referred to as the internet plumping problem.

(vii) Data analysis

Analyzing the massive data and obtaining beneficial information is a challenging task. Often, the customers desire the outcome of the analysis in high-quality formats that are simple to understand and operate. However, meeting such requirements is difficult when a high amount of heterogeneity and variety is present in the data. Data generated from different fields can be analyzed by using various tools. Choice of the tool and the analysis is very important for extracting meaningful information. Different data analysis methods used are correlation analysis, factor analysis, cluster analysis, regression analysis, memory-level analysis, massive analysis, and business intelligence analysis.

(viii) Upscaling hurdles

Data is increasing at an exponential rate. Therefore, the storage capacities of the current hardware need upscaling according to the present and future requirements. An appropriate architecture based on the forecast can be used for upscaling the hardware and avoiding difficulties in the future.

(ix) Data security

Big Data generally comprises some personal information of the users. To avoid the misuse of personal information, the data is typically anonymized before storage. For this purpose, all identifiers are removed from the data. Even with this encryption, sometimes the data is not completely anonymized. In such cases, fully homomorphic encryption (FHE) is used to anonymize the data completely. However, anonymization reduces the efficiency of data usage as it removes some part of the data. Therefore, the right balance between the efficiency of the data usage and privacy of the data is required. An organization's security model involves achieving confidentiality, integrity, and availability as the key objectives [70].

Further research is required to enhance the efficiency, analysis, and storage of Big Data. Since storage is one of the biggest bottlenecks with Big Data analysis, it inspired several researchers to look beyond the conventional data storage method to DNA-based data storage.

3. DNA computing-based Big Data storage

Although the DNA-based storage devices are entirely different from conventional data storage devices, both run parallelly at the lowest level. The traditional storage device stores the information in binary bits. Unlike this, a quaternary system comprising four nucleotide bases (A, T, G, or C) joined by a phosphodiester bond in a DNA molecule is used to store the data in DNA-based storage.

3.1 Write down to DNA

The steps performed for writing the digital data on the DNA are discussed in this section.

Step 1: In the first step, *n* binary (0 and 1) digits are converted to *n*/2 quaternary digits (0, 1, 2, and 3) (i.e., 11010001 → 3101).

Step 2: In the second step, quaternary digits (0, 1, 2, and 3) are mapped to DNA nucleotides (A, C, G, and T) (i.e., 3101 → TCAC). For instance, the binary string 11010001 represents the base four string as 3101. This base four string further means TCAC. Additionally, the probability of several experimental errors is reduced by encoding binary data in base three instead of base four. A binary string 01100001 can be mapped to the base three string as 01112 by using the Huffman code. The mapping of each ternary digit to a DNA nucleotide is based on rotating code. This string 01112 maps into DNA sequence CTCTG by rotating nucleotide encoding. It prevents the repetition of the same nucleotide twice and is extremely important as the repetition of the same nucleotide considerably enhances the chance of error in the sequencing.

Step 3: In the third step, the corresponding single-stranded DNA sequences are synthesized by using a DNA synthesizer. Using the existing DNA synthesis technology, it is not possible to synthesize large DNA sequences, and only the sequences with hundreds of bits can be synthesized with a very low error rate. Therefore, the encoding of a complete solution is difficult as it requires large-size DNA sequences to be synthesized. This challenge is handled to a large extent by segmenting the large information in multiple small DNA sequences arranged in blocks. In these blocks, each DNA strand is synthesized separately, and therefore it allows storing large values with a very low error rate.

Step 4: In step 4, the strands are assigned with some indexes so that the DNA strands tag to corresponding primers that segregate molecules of interest and thus execute random access. Since DNA sequences synthesized in a DNA pool are randomly collected to the decoder, indexing becomes very important. Amplification of each data block occurs by addressing the information to distinguish its location in the input data sharing. There are two parts to the address space. The one part of the address recognizes the key. The second part is the address index of the block which contains the value linked with the key. The combined address is represented by a fixed length of nucleotides. Typically, 24 bits are used for indexing as it gives $2^{24} = 16{,}777{,}216$ distinct indexed sequences.

Further, the encoding process is associated with errors occurring in DNA writing. Errors such as missing the nucleotide (deletion error), the addition of nucleotide when it is not required (insertion error), and the addition of wrong nucleotide (substitution error) occur at the time of synthesis. Also, a few DNA sequences are lost, or some redundant ones are generated at the time of writing, affecting the recovery of information. Some additional information is encoded in the DNA sequences to eliminate the above errors to ensure that the data recovery is perfect.

Step 5: In step 5, all the strands generated are amplified by performing the PCR. This improves the accuracy.

3.2 Read out from DNA

The steps performed for reading the DNA sequences are discussed in this section.

Step 1: In the first step, the DNA sample is drawn from the DNA pool obtained in the write-down procedure.

Step 2: In this step, the DNA sample is amplified using PCR with primers specific to intended information, and DNA of the desired length is extracted by using gel electrophoresis.

Step 3: In this step, the unused DNA templates, primer-dimers, unbound primers, and unused Taq DNA polymerase are removed from the pool of the DNA as these can decrease sequencing efficiency.

Step 4: After extraction of the DNA, two adaptors are added to the two ends of the DNA. These adaptors are the sequencing adaptors added to the DNA sequence one by one in two consecutive PCR. After each step of the PCR, agarose gel electrophoresis is performed for extracting the DNA of the desired length. Again, it removes unused DNA templates, primer-dimers, unbound primers, and unused Taq DNA polymerase from the DNA pool.

Step 5: In this step, the sample is subjected to the DNA sequencing process. Here, each DNA strand is randomly chosen from the DNA pool, and nucleotide sequences of this stand are identified. Thus, the output file generated by the sequencer has DNA sequences of millions of molecules. Since the process is stochastic, there are chances that some strands are more often read than the others. Also, there are high chances that several DNA strands remain unread, so a large quantity of samples is used to avoid this. Since more individual DNA strands are naturally readout than available DNA variants in the pool, the sequencing coverage, which is a ratio of the variants present to the total number of DNA readout, is an important factor. Typically, the required content is directly proportional to the natural logarithm of the number of diverse DNA variants present in the pool. This gives the estimation of required sequencing coverage for reading the complete information. The output of the DNA Sequencing machine is generated in the fasta format.

Step 6: In this step, the DNA sequences present in the fasta file are converted back to the binary format. Each sequence is translated back to the bits by mapping and protecting the individual sequence. There is a high possibility of obtaining multiple sequences for each index due to numerous errors. Therefore, sequences that present most frequently are chosen. This methodology operates excellently in the low-error regime for multiple reads. Organick et al. [38] proposed another method that works well for the high error regime. In this method, candidate DNA sequences for decoding are identified from the reads by first clustering the reads. This method also shows a slight advantage in the low-error regime. However, it additionally requires clustering. Unlike this, correction can be directly used in the encoding by using the additional redundant sequences. Since the errors occurring during the synthesis, storage, and sequencing yield the original sequence subset, reading of the redundant sequences is used to correct the missing sequences of real digital information. These sequences are then transformed back to the actual binary sequences. For this conversion, the same encoding used for converting the binaries to quaternary and then to A, T, G, and C sequences is reversed.

3.3 Experimental details

3.3.1 Experimental environment

A specific experimental environment is required for practically storing the information using DNA. Since all the digital data is present in binary format, it is first converted to a quaternary format using the appropriate encoding scheme. Based on this, the appropriate DNA sequences are designed. A high-performance computing device is essential for this task. The DNA sequences designed are synthesized by using the DNA synthesizer. A clean room with a controlled temperature requires the experimental process to avoid contamination and degradation. Further, the DNA sequences can be stored at 4 °C for 2–4 weeks, at −20 °C, or −80 °C for a longer storage period. DNA sequencing is used to retrieve and read the information, which generates the output in fasta format. These DNA sequences are then decoded back to binary form using the computer program.

3.3.2 Experimental steps

Fig. 3 shows the experimental procedure of DNA-based data storage. Four major experimental steps are typically involved in DNA data storage. These steps are writing, storage, retrieval, and reading.

(i) Writing

First, the digital data is converted to DNA sequences by using the appropriate coding. This coding scheme may vary as per the user. The coding and decoding should be the same to get an accurate result. Few properties and parameters such as termination level, bits encoded per base, and possible length of the sequences are calculated using the appropriate code. This code takes the digital data file as the input and gives a text file containing the DNA sequences as the output file.

(ii) Storage

The DNA sequences generated are synthesized in the writing step using a DNA synthesizer. These DNA sequences are highly concentrated. Therefore, DNA sequences are diluted to a concentration of 5 ng/μL using PCR-grade water. This stock solution is stored at 4 °C for 2–4 weeks, at −20 °C or −80 °C for a longer storage period. Next, the PCR is performed on the synthesized DNA sequences three times. In the first PCR, the desired DNA sequences generated are amplified for data storage. Then the PCR product is separated using gel electrophoresis for 120–170 nucleotides based on the payload. There are some possibilities of loss of the sample due to the removal of short oligonucleotides. Therefore, it is required to check the

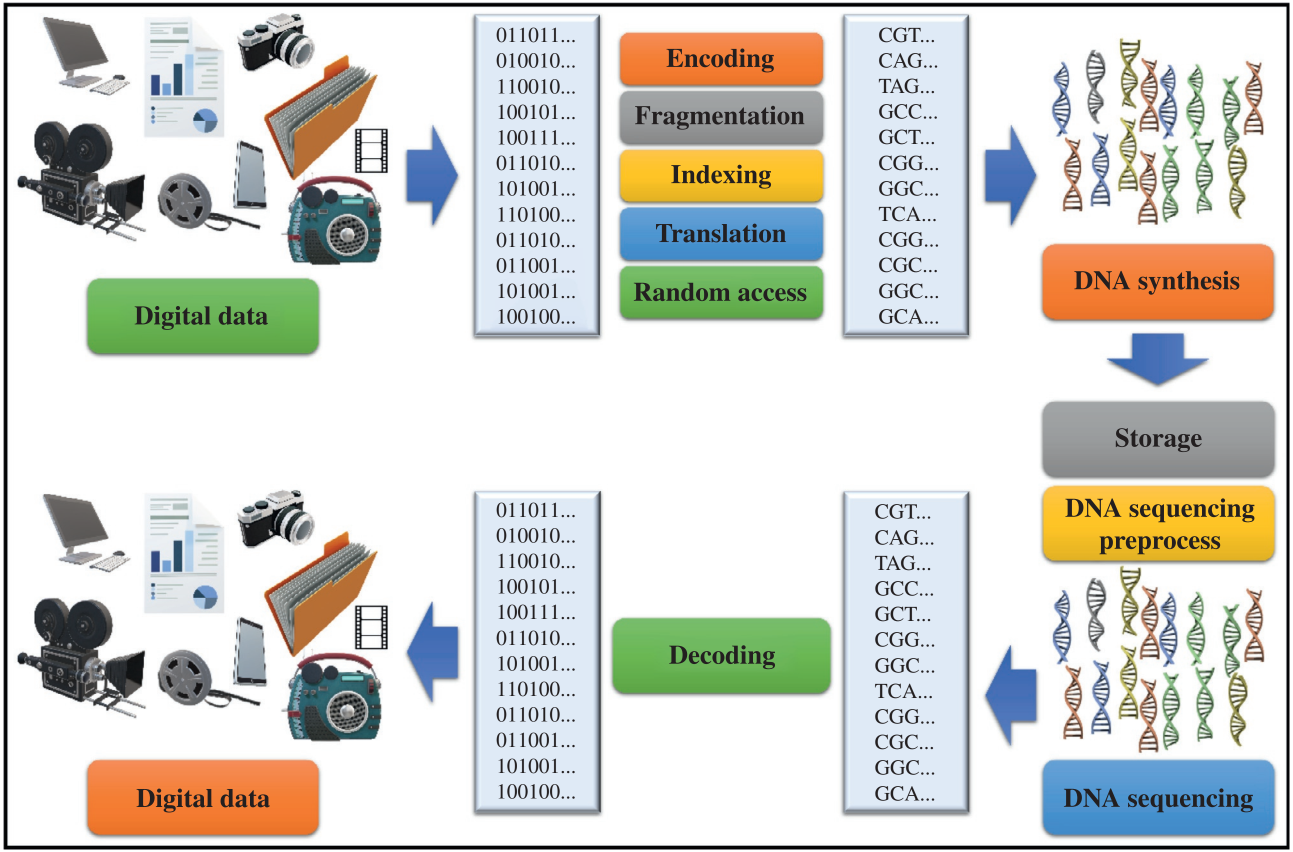

Fig. 3 Process flow of the DNA-based storage system.

quality of output using fluorescence-based concentration measurement methods. Then, the second PCR is performed for adding the sequencing adaptors at one end of the DNA. The PCR product is then separated using gel electrophoresis to extract an oligonucleotide band of 150–200 nucleotides. The third PCR is performed on these separated DNA sequences to add the sequencing adaptors at the remaining end of the DNA. Finally, the DNA sequences of 200–250 nucleotides are separated using gel electrophoresis. Tables 3 and 4 show the detailed parameters used in PCR.

(iii) Retrieve

The sequencing coverage is selected based on the quality of DNA synthesis and PCR cycles for retrieving the information. Conventionally, the number of initially written DNA sequences becomes almost 200 times after the reading. Illumina iSeq system or any other DNA sequencer is used to read the DNA sequences. The sequencing reads are calculated by multiplying sequencing coverage with the number of sequences encoded.

(iv) Reading

DNA sequences are obtained as a fasta file with all information of the reads and the quality control data. This file works as input data to the computer code, which decodes the DNA sequences to binary. The accuracy is calculated by comparing the original binary file with the decoded binary file.

Table 3 Typical quantities used in PCR.

Reagent	Amount (μL in 1 well)	Final concentration
PCR master mix	10	1 ×
PCR-grade water	7	—
Primer 1F (10 μM)	1	0.5 μM
Primer 1R (10 μM)	1	0.5 μM
DNA stock solution	1	~0.005 ng per μL
Total	20	—

Table 4 Typical cycling parameters used in PCR.

Cycle number	Denaturation (temperature, duration)	Annealing (temperature, duration)	Extension (temperature, duration)
1	95 °C, 5 min	—	—
2–15	95 °C, 15 s	54 °C, 30 s	72 °C, 30 s

4. Challenges for DNA-based data storage

There are several challenges in using DNA for Big Data storage and analysis. The discussion on the significant challenges is given next.

(i) Spatial segregation

The size of the data that can be stored is limited mainly by the expense of DNA synthesis and DNA sequencing. In recent years, researchers are putting their efforts into improving the extent of DNA synthesis and encoding strategies. The increase in file index labeling significantly reduces the number of nucleotide bits available for data storage as the length of the DNA synthesized is limited. For instance, for a data set of exabyte-size, at least three barcodes are required for indexing each file [30]. Three barcodes acquire up to 60 nucleotides in each DNA sequence, ultimately reducing the number of nucleotides available for encoding the data [71].

(ii) Errors in DNA synthesis

The DNA synthesis process is carried out by attaching the DNA molecule to a solid surface, and each nucleotide is added one after another through a chemical process. Current DNA synthesis methods generate millions of DNA strands per chip. Secondly, various errors occur at the time of the extension (i.e., new nucleotide addition) of the DNA strand. These errors are (a) *Deletion* (i.e., the attachment of the desired nucleotide is not at its position), (b) *Insertion* (i.e., a wrong nucleotide is attached at a position somewhere it should not be), and (c) *Substitution* (i.e., the addition of a nucleotide other than the expected). In some cases, it might be possible that the distribution of the DNA generated at each spot on a solid surface is uneven [17,38].

(iii) Errors in DNA storage

The DNA processing and DNA data storage steps such as removing DNA from the solid surface and heating cycles of PCR may result in DNA decay. The significant consequences of damage are depurination which ultimately results in DNA strand break down. The PCR amplification of these inadequate DNA cannot be performed as primers are absent on both ends of DNA strands. This may lead to substitution errors and considerable DNA loss at the time of storage. Error-correcting codes can be employed for the correction of storage-related errors [28,33,72].

(iv) Errors in DNA sequencing

DNA sequencing errors depend on the methodology used for the sequencing. Illumina is the most widely used DNA sequencing method. The error rates of Illumina are higher at the end of a read as these errors are strand-specific. The substitution reading error rates depend on the dataset. An exact number is about 0.0015–0.0004 errors per base for substitution and 10^{-6} for insertions and deletion errors [17,32,38].

(v) Data redundancy

Duplication of the data arises if two or more data samples represent the same object. Data inconsistency and duplication severely affect storage. Although many techniques are available to detect duplicate files, they are not feasible for Big Data storage using DNA [30,31].

(vi) Data noise

In DNA-based data storage systems, data noise is created at different stages of the process (e.g., substitution, deletion, and insertion errors on the DNA sequences create unwanted DNA sequences). In some cases, homopolymers and hairpin formation also generate noisy data. These unwanted DNA sequences generate noisy reads at the time of DNA sequencing.

(vii) The time required for retrieving the information

The DNA synthesis and DNA sequencing are mostly automated, but some intermediate steps are performed manually. A fully automated DNA-based storage device is developed and demonstrated by Takahashi et al. [32]. Each process step requires some time. Also, preparation of the experiment takes some time. The overall time required for automatic DNA data storage system is approximately 21 h.

(viii) Extending to Big Data analytics

Though several researchers have illustrated DNA-based data storage, the processing and analytics of Big Data using DNA are still awaited. Unlike the conventional tools developed for Big Data analytics, such DNA-based processing tools are yet to develop.

5. Conclusions and future work

High endurance and massive information density make DNA a suitable medium for data storage. Although DNA is known for storing biological information, storing digital information is challenging. Recent progress of storing data up to 200 MB using DNA opens new prospects for handling Big Data storage challenges. Particularly, low energy requirements and long storage times offered by DNA systems can be extremely useful in storing important information for a much-extended period than conventional storage devices. Also, DNA-based computers' progress for solving a variety of combinatorial and computing problems opens the possibility of developing

DNA-based tools for Big Data analytics. Developing DNA-based data security, making DNA-based storage more convenient for the nonexpert user, and applying DNA-based storage for real-time Big Data storage are some of the open problems in this area.

Acknowledgment

The partial financial support from the Science and Engineering Research Board, Government of India, New Delhi (through grant EEQ/2020/000460, dated December 18, 2020) is gratefully acknowledged.

References

[1] Y. Cevallos, et al., On the efficient digital code representation in DNA-based data storage, in: *Proceedings of the 7th ACM International Conference on Nanoscale Computing and Communication, NanoCom 2020*, 2020, https://doi.org/10.1145/3411295.3411314.

[2] A. Agrahari, D.T.V.D. Rao, A review paper on big data: technologies, tools and trends, Int. Res. J. Eng. Technol. (2017) 640–649.

[3] J. Manyika, et al., Big Data: The Next Frontier for Innovation, Competition, and Productivity, McKinsey Global Institute, 2011.

[4] C.K. Emani, N. Cullot, C. Nicolle, Understandable big data: a survey, Comput. Sci. Rev. 17 (2015) 70–81, https://doi.org/10.1016/j.cosrev.2015.05.002.

[5] A. Oussous, F.Z. Benjelloun, A. Ait Lahcen, S. Belfkih, Big data technologies: a survey, J. King Saud Univ. Comput. Inf. Sci. 30 (4) (2018) 431–448, https://doi.org/10.1016/j.jksuci.2017.06.001.

[6] J. Wang, Y. Yang, T. Wang, R. Simon Sherratt, J. Zhang, Big data service architecture: a survey, J. Internet Technol. 21 (2) (2020) 393–405, https://doi.org/10.3966/160792642020032102008.

[7] T.R. Rao, P. Mitra, R. Bhatt, A. Goswami, The big data system, components, tools, and technologies: a survey, Knowl. Inf. Syst. 60 (3) (2019). Springer London.

[8] R. Urwongse, K. Culver, Applications of Blockchain in Healthcare, Springer Nature, 2021.

[9] L. Zhou, S. Pan, J. Wang, A.v. Vasilakos, Machine learning on big data: opportunities and challenges, Neurocomputing 237 (January) (2017) 350–361, https://doi.org/10.1016/j.neucom.2017.01.026.

[10] M. Chen, S. Mao, Y. Zhang, V.C.M. Leung, Big data related technologies, challenges and future prospects, Inf. Technol. Tour. 15 (3) (2015) 283–285, https://doi.org/10.1007/s40558-015-0027-y.

[11] S. Ullah, M.D. Awan, M. Sikander Hayat Khiyal, Big data in cloud computing: a resource management perspective, Sci. Program. 2018 (2018), https://doi.org/10.1155/2018/5418679.

[12] I. Anagnostopoulos, S. Zeadally, E. Exposito, Handling big data: research challenges and future directions, J. Supercomput. 72 (4) (2016) 1494–1516, https://doi.org/10.1007/s11227-016-1677-z.

[13] M. Wada, N. Tanaka, Big data: survey, technologies, opportunities, and challenges Nawsher, Jpn. J. Appl. Phys. 29 (8) (1990) L1497–L1499, https://doi.org/10.1143/JJAP.29.L1497.

[14] H. Valladas, et al., Radiocarbon AMS dates for Paleolithic cave paintings, Radiocarbon 43 (2B) (2001) 977–986.

[15] W. Kutschera, W. Rom, Ötzi, the prehistoric Iceman, Nucl. Instrum. Methods Phys. Res. Sect. B Beam Interact. Mater. Atoms 164 (2000) 12–22, https://doi.org/10.1016/S0168-583X(99)01196-9.

[16] A. Keller, et al., New insights into the Tyrolean Iceman's origin and phenotype as inferred by whole-genome sequencing, Nat. Commun. 3 (2012), https://doi.org/10.1038/ncomms1701.

[17] L. Ceze, J. Nivala, K. Strauss, Molecular digital data storage using DNA, Nat. Rev. Genet. 20 (8) (2019) 456–466, https://doi.org/10.1038/s41576-019-0125-3.

[18] M.G.T.A. Rutten, F.W. Vaandrager, J.A.A.W. Elemans, R.J.M. Nolte, Encoding information into polymers, Nat. Rev. Chem. 2 (11) (2018) 365–381, https://doi.org/10.1038/s41570-018-0051-5.

[19] M. Kunder, Daily estimated size of the world wide web. https://www.worldwidewebsize.com/.

[20] P.Y. De Silva, G.U. Ganegoda, New trends of digital data storage in DNA, Biomed. Res. Int. 2016 (2016) 1–14, https://doi.org/10.1155/2016/8072463.

[21] S. Shrivastava, R. Badlani, Data storage in DNA, Int. J. Electr. Energy 2 (2) (2014) 119–124, https://doi.org/10.12720/ijoee.2.2.119-124.

[22] M. Blawat, et al., Forward error correction for DNA data storage, Procedia Comput. Sci. 80 (2016) 1011–1022, https://doi.org/10.1016/j.procs.2016.05.398.

[23] M.S. Neiman, Some fundamental issues of microminiaturization, Radiotekhnika 1 (1) (1964) 3–12.

[24] H. Lodish, et al., Molecular Cell Biology, seventh ed., W. H. Freeman, 2012.

[25] D.L. Nelson, A.L. Lehninger, M.M. Cox, Lehninger Principles of Biochemistry, Macmillan, 2008.

[26] J.D. Watson, F.H.C. Crick, Molecular structure of nucleic acids, Nature 171 (4356) (1953) 737–738.

[27] K.S. Gates, An overview of chemical processes that damage cellular DNA: spontaneous hydrolysis, alkylation, and reactions with radicals, Chem. Res. Toxicol. 22 (11) (2009) 1747–1760.

[28] R.N. Grass, R. Heckel, M. Puddu, D. Paunescu, W.J. Stark, Robust chemical preservation of digital information on DNA in silica with error-correcting codes, Angew. Chem. Int. Ed. 54 (8) (2015) 2552–2555, https://doi.org/10.1002/anie.201411378.

[29] A.K.-Y. Yim, et al., The essential component in DNA-based information storage system: robust error-tolerating module, Front. Bioeng. Biotechnol. 2 (2) (2014) 1–5, https://doi.org/10.3389/fbioe.2014.00049.

[30] J. Bornholt, et al., A DNA-based archival storage system, IEEE Micro 44 (2) (2017) 637–649, https://doi.org/10.1109/MM.2017.264163456.

[31] R. Laddha, K. Honwadkar, Digital data storage on DNA, Int. J. Comput. Appl. 142 (2) (2016) 43–46.

[32] C.N. Takahashi, B.H. Nguyen, K. Strauss, L. Ceze, Demonstration of end-to-end automation of DNA data storage, Sci. Rep. 9 (1) (2019) 1–5, https://doi.org/10.1038/s41598-019-41228-8.

[33] R. Lopez, et al., DNA assembly for nanopore data storage readout, Nat. Commun. 10 (1) (2019) 2933, https://doi.org/10.1038/s41467-019-10978-4.

[34] L.C. Meiser, et al., Reading and writing digital data in DNA, Nat. Protoc. 15 (1) (2020) 86–101, https://doi.org/10.1038/s41596-019-0244-5.

[35] N. Goldman, et al., Toward practical high-capacity low-maintenance storage of digital information in synthesised DNA, Nature 494 (7435) (2013) 77–80, https://doi.org/10.1038/nature11875.

[36] G.M. Church, Y. Gao, S. Kosuri, Next-generation digital information storage in DNA, Science 337 (6102) (2012) 1628, https://doi.org/10.1038/21092.

[37] Y. Erlich, D. Zielinski, DNA Fountain enables a robust and efficient storage architecture, Science 355 (6328) (2017) 950–954, https://doi.org/10.1126/science.aaj2038.

[38] L. Organick, et al., Random access in large-scale DNA data storage, Nat. Biotechnol. 36 (3) (2018) 242–248, https://doi.org/10.1038/nbt.4079.

[39] S. Namasudra, R. Chakraborty, A. Majumder, N.R. Moparthi, Securing multimedia by using DNA-based encryption in the cloud computing environment, ACM Trans. Multimed. Comput. Commun. Appl. 16 (3s) (2021) 1–19, https://doi.org/10.1145/3392665.

[40] S. Namasudra, Fast and secure data accessing by using DNA computing for the cloud environment, IEEE Trans. Serv. Comput. (2020), https://doi.org/10.1109/TSC.2020.3046471.

[41] S. Namasudra, S. Sharma, G.C. Deka, P. Lorenz, DNA computing and table based data accessing in the cloud environment, J. Netw. Comput. Appl. 172 (April) (2020), 102835, https://doi.org/10.1016/j.jnca.2020.102835.

[42] S. Namasudra, An improved attribute-based encryption technique towards the data security in cloud computing, Concurr. Comput. Pract. Exp. 31 (3) (2019) e4364, https://doi.org/10.1002/cpe.4364.

[43] L. Anavy, I. Vaknin, O. Atar, R. Amit, Z. Yakhini, Improved DNA based storage capacity and fidelity using composite DNA letters, bioRxiv (2018) 433524, https://doi.org/10.1101/433524.

[44] Y. Choi, et al., Addition of degenerate bases to DNA-based data storage for increased information capacity, bioRxiv (2018) 367052, https://doi.org/10.1101/367052.

[45] S.M.H.T. Yazdi, R. Gabrys, O. Milenkovic, Portable and error-free DNA-based data storage, Sci. Rep. 7 (1) (2017) 1–6, https://doi.org/10.1038/s41598-017-05188-1.

[46] H.H. Lee, R. Kalhor, N. Goela, J. Bolot, G.M. Church, Enzymatic DNA synthesis for digital information storage, bioRxiv (2018) 348987, https://doi.org/10.1101/348987.

[47] L.M. Adleman, Molecular computation of solutions to combinatorial problems, Science 266 (5187) (1994) 1021–1024.

[48] R.J. Lipton, DNA solution for hard computational problems, Science 268 (5210) (1995) 542–545.

[49] L.M. Smith, et al., A surface-based approach to DNA computation, J. Comput. Biol. 5 (2) (1998) 255–267.

[50] Q. Liu, L. Wang, A.G. Frutos, A.E. Condon, R.M. Corn, L.M. Smith, DNA computing on surfaces, Nature 403 (6766) (2000) 175–179.

[51] R.S. Braich, N. Chelyapov, C. Johnson, P.W.K.K. Rothemund, L.M. Adleman, Solution of a 20-variable 3-SAT problem on a DNA computer, Science 296 (5567) (2002) 499–502.

[52] D. Faulhammer, A.R. Cukras, R.J. Lipton, L.F. Landweber, Molecular computation: RNA solutions to chess problems, Proc. Natl. Acad. Sci. U. S. A. 97 (4) (2000) 1385–1389.

[53] Q. Ouyang, P.D. Kaplan, S. Liu, A. Libchaber, DNA solution of the maximal clique problem, Science 278 (5337) (1997) 446–449.

[54] K. Sakamoto, et al., Molecular computation by DNA hairpin formation, Science 288 (5469) (2000) 1223–1226.

[55] D. Sharma, M. Ramteke, A note on short-term scheduling of multi-grade polymer plant using DNA computing, Chem. Eng. Res. Des. 135 (2000) (2018) 78–93.

[56] J. Chao, et al., Solving mazes with single-molecule DNA navigators, Nat. Mater. 18 (3) (2019) 273–279.

[57] D. Sharma, M. Ramteke, In vitro identification of the Hamiltonian cycle using a circular structure assisted DNA computer, ACS Comb. Sci. 22 (5) (2020) 225–231, https://doi.org/10.1021/acscombsci.9b00150.

[58] D. Sharma, M. Ramteke, DNA computing: methodologies and challenges, in: DNA- and RNA-Based Computing Systems, Wiley Online Library, 2021, pp. 15–29.

[59] W. Fan, A. Bifet, Mining big data: current status, and forecast to the future, ACM SIGKDD Explor. Newslett. 1 (1) (2013) 1–5, https://doi.org/10.21742/ijpccem.2014.1.1.01.

[60] A. McAfee, E. Brynjolfsson, T.H. Davenport, D. Patil, D. Barton, Big data: the management revolution, Harv. Bus. Rev. 90 (10) (2012) 60–68.

[61] V. Rajaraman, Big data analytics, Resonance 21 (8) (2016) 695–716.

[62] A. Gandomi, M. Haider, Beyond the hype: big data concepts, methods, and analytics, Int. J. Inf. Manage. 35 (2) (2015) 137–144, https://doi.org/10.1016/j.ijinfomgt.2014.10.007.

[63] R. Ratra, P. Gulia, Big data tools and techniques: a roadmap for predictive analytics, Int. J. Eng. Adv. Technol. 9 (2) (2019) 4986–4992, https://doi.org/10.35940/ijeat.B2360.129219.

[64] A. Mohan, Big data analytics: recent achievements and new challenges, Int. J. Comp. Appl. Technol. Res. 5 (7) (2016) 460–464.

[65] M. Hussain Iqbal, T. Rahim Soomro, Big data analysis: Apache Storm perspective, Int. J. Comput. Trends Technol. 19 (1) (2015) 9–14.

[66] B. Thillaieswari, Comparative study on tools and techniques of big data analysis, Int. J. Adv. Netw. Appl. 8 (5) (2017) 61–66.

[67] M.S. Al-Hakeem, A proposed big data as a service (BDaaS) model, Int. J. Comput. Sci. Eng. 4 (11) (2016) 15–21.

[68] A. Lakshman, M. Prashant, Cassandra: a decentralized structured storage system, ACM SIGOPS Oper. Syst. Rev. 44 (2) (2010) 35–40.

[69] D. Featherston, Cassandra: Principles and Application, 2010.

[70] S. Namasudra, D. Devi, S. Choudhary, R. Patan, S. Kallam, Security, privacy, trust, and anonymity, in: Advances of DNA Computing in Cryptography, Taylor & Francis, 2018, pp. 138–150.

[71] J.L. Banal, T.R. Shepherd, J. Berleant, H. Huang, M. Reyes, Random access DNA memory using Boolean search in an archival file storage system, Nat. Mater. 20 (9) (2021) 1272–1280, https://doi.org/10.1038/s41563-021-01021-3.

[72] K. Chen, E. Winfree, Error correction in DNA computing: misclassification and strand loss, Computing (2000) 49–63.

About the authors

Manojkumar Ramteke received a B. Tech degree in Chemical Engineering from Dr. Babasaheb Ambedkar Technological University, Lonere (MH), M. Tech, and Ph. D. in Chemical Engineering from the Indian Institute of Technology, Kanpur. Dr. Ramteke is an Associate Professor at the Department of Chemical Engineering, Indian Institute of Technology Delhi. His current research interests are industrial process modeling and multi-objective optimization, scheduling, planning and control of process operations, metaheuristic algorithms, and novel computing methods.

Deepak Sharma received a B. Tech from Maharshi Dayanand University, Rohtak, and M. Tech. from the Indian Institute of Technology Roorkee. He acquired a Ph.D. from the Indian Institute of Technology Delhi, New Delhi, India. He also worked as a Research Associate at the Department of Chemical Engineering, Indian Institute of Technology Delhi, on Deep Learning algorithms for early diagnosis of pancreatic cancer and cardiovascular disease.