Analysis of security and privacy challenges for DNA-genomics applications and databases

Saadia Arshad[a], Junaid Arshad[b,*], Muhammad Mubashir Khan[a] and Simon Parkinson[c]

[a]Department of Computer Science & IT, NED University of Engineering and Technology, Karachi, Pakistan
[b]School of Computing and Digital Technology, Birmingham City University, Birmingham, UK
[c]Department of Computer Science, University of Huddersfield, Huddersfield, UK

ARTICLE INFO

Keywords:
Cyberbiosecurity
DNA
Genomics
Cyber-attacks
Vulnerabilities
Bioinformatics

ABSTRACT

DNA technology is rapidly moving towards digitization. Scientists use software tools and applications for sequencing, synthesizing, analyzing and sharing of DNA and genomic data, operate lab equipment and store genetic information in shared datastores. Using cutting-edge computing methods and techniques, researchers have decoded human genome, created organisms with new capabilities, automated drug development and transformed food safety. Such software applications are typically developed to progress scientific understanding and as such cyber security is never a concern for these applications. However, with the increasing commercialisation of DNA technologies, coupled with the sensitivity of DNA data, there is a need to adopt a security-by-design approach. In this paper we investigate bio-cyber security threats to genomic-DNA data and software applications making use of such data to advance scientific research. Specifically, we adopt an empirical approach to analyse and identify vulnerabilities within genomic-DNA databases and bioinformatics software applications that can lead to cyber-attacks affecting the confidentiality, integrity and availability of such sensitive data. We present a detailed analysis of these threats and highlight potential protection mechanisms to help researchers pursue these research directions.

1. Introduction

A DNA or deoxyribonucleic acid is the hereditary material in almost all living organisms. It is made up of four coded chemical bases denoted as A, T, C, and G. The sequence of these chemical bases indicates the information available for building and maintaining an organism. A genome – an individual's entire DNA set – is the representation of an individual's personal, biological characteristics and is therefore sensitive data [12].

A person's DNA or genome is unique and is considered private or personal information, containing information regarding one's family and ancestors. It is useful in uniquely identifying an individual's hereditary traits such as inherited diseases, adverse reactions to common drugs, genetic predisposition etc. The use of genomic information is global and so is the importance of providing adequate protection. According to US national library of medicine, a genome contains complete information required to build and maintain that organism [46], and the EU General Data Protection Regulation (GDPR) has emphasised that there is a need to take great care when handling genomic information [16].

DNA technology is promptly moving towards digitization. Using cutting-edge computing technologies and software programs, researchers are decoding human genome, designing new drugs, and writing modified DNA code. The 1,000 genomes project is a prominent example of initiatives that are using innovative computer technologies to establish the most detailed catalogue of human genetic variation [22]. Scientists frequently use contemporary computing technologies and methods to conduct everyday tasks such as, uploading genomes onto online databases, analyzing genomic-DNA data, operating lab equipment, running standard bioinformatics processes, and sharing data among organizations, researchers, clinicians, and individual users. Alongside the extraordinary benefits brought by digitization of DNA technology, it introduces concerns with respect to security of data and processes used by scientists. Since genomic data contains information about a family; the impact of a breach of such data also affects the person's close and distant biological relatives [15], making it more significant as compared to attributes such as name, date of birth and address of an individual.

Within this context, this paper is focused at conducting an in-depth security assessment of software used within bioinformatics that process genomic/DNA data and databases. Specifically, the paper uses in-depth analysis of existing literature to develop a taxonomy of common security and privacy issues in bioinformatics tools and databases. This is envisaged to improve awareness within the bioinformatics community with respect to potential vulnerabilities and threats for data and processes used for cutting-edge research. Furthermore, the paper uses static code analysis techniques to analyse popular software for genome analysis, highlighting vulnerabilities that exist in these tools. It is crucial to address these vulnerabilities because, if left untreated, they could lead to severe bio-cyber security attacks that can exploit such critical systems and compromise the confidentiality, integrity and availability of DNA-genomic tools and databases. This critical data can be misused to blackmail, to deny medical treatment or even in genetic warfare or bio-terrorism [23].

The paper has been constructed to provide useful insights

*Corresponding author: Junaid Arshad, School of Computing and Digital Technology, Birmingham City University, Birmingham, UK Email: junaid.arshad@bcu.ac.uk
ORCID(s):

to researchers in Bioinformatics on the vulnerabilities and potential solutions that exist in current widely used software tools. The purpose of the manuscript is also to increase awareness within the Bioinformatics community of cyber security and to ensure that cyber security risks are fully understood, especially due to the sensitive nature of DNA data. However, as cyber security is a large aspect of Computing, the manuscript deliberately omits some background and introductory content to some of the presented concepts.

The rest of the paper is organized as follows: Section 2 highlights the rationale of this research by identifying the need to protect genomic and DNA data and software used to process these. Section 3 provides a background for cyberbiosecurity and defines it as a multidisciplinary concept. Section 4 presents a critical review of the existing efforts using a systematic approach as well as defining the scope of this study. Section 5 presents details of the security analysis conducted including details of the software considered and methods used to evaluate them. Section 6 presents the taxonomy of security and privacy issues in bioinformatics tools and databases which includes definition of common vulnerabilities identified through our analysis, potential attacks and mapping between them. Section 7 presents a mapping between vulnerabilities, attacks and defence mechanisms whilst including a focused discussion on specific defence mechanisms. Section 8 concludes the paper.

2. Need to protect genomic and DNA data

Since the beginning of Human Genome Project [31] in 2003, genomic and DNA research has seen consistently remarkable progress. Due to the progress made within the field of genomics, new discoveries are being made on the daily basis such as revolutionary life saving discoveries for precision medical care. Today, the total cost and time required to generate an entire human genome sequence has dropped significantly and the use of large scale genome sequencing has established with applications in health care, drug development, and forensics investigations.

The sequencing of the human genome holds many benefits, such as the linking of an individual's genomic information and health information, coupled with advances in computing and informatics signals a new era of molecular medicine [47]. It also helps in the understanding of new viruses and their treatment, identification of mutations that are linked to different forms of cancer, drug development, criminal investigators in forensics, and the development of biofuels. Many countries such as the UK [30], the US [54] and Saudi Arabia [39] have efforts underway to sequence genomes of their citizens such as patients with rare diseases. With this rate of growth it is estimated that by 2025 approximately 1 billion human genomes will be sequenced [68].

In order to facilitate research progress within this discipline, sharing of genomic data among scientists, research groups and health experts is a common practice. Most study groups collect genomic data from large cohorts of individuals, including both healthy and diseased individuals. Furthermore, a common trend in recent years is direct-to-consumer (DTC) where participants sending samples of DNA to testing companies such as 23andMe, MyHeritage, and Ancestry.com to learn about their genetic ancestors and disorders. DTC companies collect large amounts of genomic data, for example GEDmatch is a popular direct-to-consumer company founded in 2010 has more than 1.3 Million individuals genomic data. Furthermore, a study by Massachusetts Institute of Technology (MIT) found out that by the start of 2019 more than 26 million customers had added their DNA information into the online databases that are being maintained by the top four DNA testing companies [64].

However, The rise in direct-to-consumer (DTC) companies introduces new security and privacy concerns when it comes to an individual's genomic data. Firstly, the DTC companies may share genomic data with third parties with minimal or no information provided to individuals before their data is used in independent research projects which is also a privacy concern. In this respect, a recent study [14] reported around 67% of DTC companies to have provided inadequate information about where they utilize collected genomic information of an individual. Secondly, Privacy breaches can have serious adverse effects on genomic or DNA driven researches as individuals will hesitate participating in such studies. Therefore, it is crucial to ensure privacy and security of genomic data. Finally, as DTC companies collect large amount of DNA data, these companies become a lucrative target for cyber-criminals.

An individual's genomic data can be highly valuable due to its significance for broad applications such as paternity testing, studying the implication of adverse reactions to common drugs, ancestry tracing, disease screening, and understanding the ways that genetic variation contributes to health and disease. The persistent concern for those who take part in genomics studies or research is that their DNA or genomic information may be used against them. For instance, life insurers might want to use this information for underwriting and to charge the correct premium [6]. Furthermore, it could be used to receive or deny medical treatment or law enforcement agencies might want to use the information to identify victims or criminal suspects [15]. Therefore, the security of such data stores in paramount.

In order to store digitized genomic data and DNA sequence, online databases are used such as GenBank [50] developed by U.S. National Center for Biotechnology Information, UCSC Genome Browser [74] and Ensembl [24] etc. As the data is difficult to interpret, various computer programs and bioinformatics tools are used to analyze and process genomic data. Therein, the use of digital technologies have increased the risk to DNA and genomic data with respect to threats such as social engineering and exploits, genomic dossiers, DNA theft, genomic data theft, and genetic warfare. For instance, a famous DNA testing service was breached by cybercriminals [10] in 2017 where hackers were understood to be able to breach 92 million accounts. Although the attackers only accessed encrypted ID and passwords, and the original DNA or genetic data remained safe,

the incident highlighted the need to analyse and understand security requirements of DNA information and to develop bespoke solutions to address these requirements.

Additionally, breach of such data could also be used to mask a genetic condition or create bio-genomic weapons for bio-terrorism [23, 75]. For instance, recent advances in genetic engineering such as CRISPR (gene editing technology) have allowed the insertion of artificial or man-made DNA strands into the living cells of organisms [2]. Although these applications have been developed to explore new cures to genetic diseases such as sickle cell anemia [42], if in wrong hands such applications can prove to be destructive. For example, this tool can be used to edit and create *designer babies* [69] or mutated animals [79]. Therefore, the security or methods and mechanisms facilitating genomic data analysis is paramount to avoid potential data breaches.

Acts and regulations designed to protect genomic data provide vague directions. For instance, healthcare organizations in the US are bound to comply with HIPAA privacy rule. The privacy rule was designed to ensure that the health information of a person is protected without delaying the information flow that is crucial in provide high-quality health care. This rule defines protected health information (PHI) such as an individual's name and social security number (SSN). [45]. However, even with protected PHI, re-identification attacks are possible. Similarly, the Genetic Information Nondiscrimination Act (GINA) of 2008 protects an individuals genetic information from discrimination from health insurers etc. but it does not clarify the extent of information that needs protection and how this data protection is carried out. Furthermore, current studies highlight that a number of countries around the globe have minimal or no regulations when it comes to genetic data protection.

3. Cyberbiosecurity

The term *cyberbiosecurity* was coined in 2018 and is defined as a collective mixture of multiple disciplines such as cyber security, bio-security and cyber physical security [49, 73] as demonstrated in Figure 1. The aim of defining this trans-disciplinary field is to understand the vulnerabilities that can occur at the interface of multiple disciplines, biomedical systems, bioinformatics tools [49] and to develop measures that will mitigate these vulnerabilities protecting important personally identifiable data from threats targeting individuals or organizations.

In recent years, a number of cyber-attacks have been reported aimed at compromising the security of biological systems. Ransomware attacks such as [51] are an example of such breaches motivating researchers to focus on developing methods to secure and protect health records, DNA and genomic data. But the importance of understanding the need of cybersecurity is still lacking in the biological research and health care industries [40].

In the past, significant effort has been made to raise awareness regarding the consequences of a lack of protection for DNA and genomic data, but researchers are still using sys-

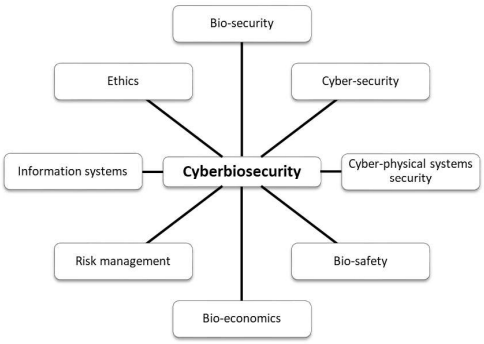

Figure 1: Different disciplines that might contribute to cyberbiosecurity as a new trans-discipline

tems that seem to lack sufficient security controls. Use of vulnerable software is one of major causes of successful cyber-attacks, therefore it is necessary to identify and protect against such vulnerabilities so as to facilitate bioinformatics research whilst achieving the required level of protection. Cyberbiosecurity aims to fill this gap by taking a multidisciplinary approach to enhance understanding of the security risks, particularly due to the increased use of information technology in the field of life sciences or medical sciences. However, identifying existing vulnerabilities in these tools that deal with DNA/genomic data, web applications and databases is essential but non-trivial task and therefore requires extensive effort. Our research is focused at addressing this challenge so as to improve the state of the art within cyberbiosecurity. In this paper, we have used static code analysis approach to discover vulnerabilities in commonly used open-source software programs, tools or databases that process or store DNA and genomic data. The tools that are used have wide-scale use in the cyber secuirty discipline and therefore can be regarded as reliable. As such, our analysis is agnostic of the programming language and therefore we have considered software written in languages such as C, C++, and JAVA.

4. Related works and scope of study

In this section, we present our efforts to identify and analyse existing works related to this paper and define the scope of this study. In particular, we highlight the methodology adopted to identify and review existing literature, a summary of prominent existing efforts, and the objectives of our study with respect to security of bioinformatics datasets and software.

4.1. Review methodology

In order to assess the state of the art within this topic, we conducted a systematic study to identify and analyse existing work within cyberbiosecurity especially emphasising works focused at security and privacy of DNA and genomic data. A summary of the keywords used along with the number of results for each keyword are presented in Table 1 whereas the overall method adopted to conducted this study is pre-

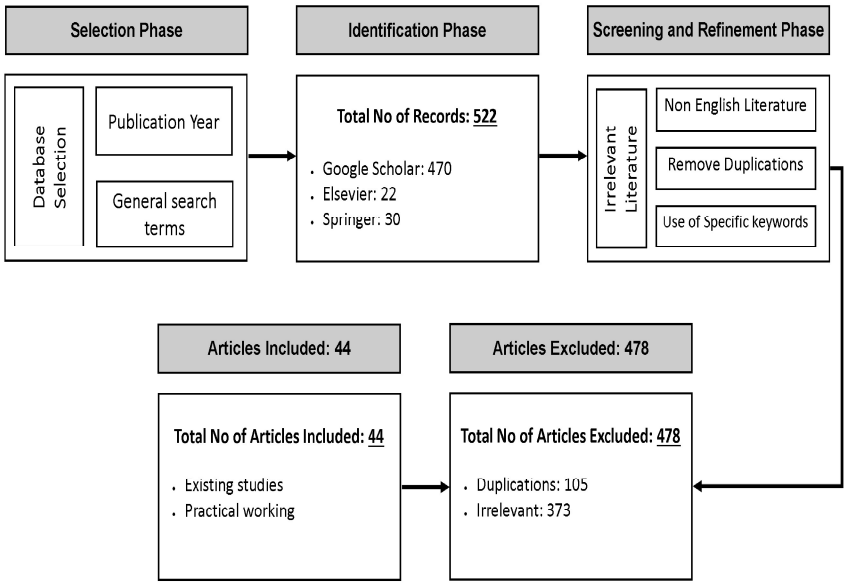

Figure 2: Systematic survey method

sented in Fig 2 which highlights different stages involved in the study.

After analysing each and every result obtained from our keyword search we eliminated multiple papers as shown in Figure 1, and ended up having 44 articles that were closely related or somehow can aid our work. To exclude and include research articles we have used a systematic survey method. Our method consisted of three phases:

1. Selection;
2. Identification; and
3. Screening and refinement;

In the **selection phase**, we used general keywords, publication year and the platform to search for our required research work which in our case was Google scholar, Elsevier and Springer since they are widely used. In the **identification phase**, we collected the search result generated from our selected keywords. The total number of results generated from our queries was 522. In the **screening and refinement phase**, we further refined our findings by removing duplicate results, irrelevant literature and non-English literature from our search result. After screening and refining, total number of research articles that were excluded were 478, we eliminated 105 duplicated articles and 373 irrelevant articles with only 44 articles remaining. All the remaining articles were focused on cyberbiosecurity and also the privacy and security of genomic and DNA data. The result of literature analysis is shown in Figure 2.

4.2. Existing efforts to analyse security of bioinformatics tools and genome databases

Recent biological researches rely on publicly available genome databases [8] and different software tools. Many of the software tools that are being used by bioinformaticians for sequencing (reading), synthesizing (writing), analyzing, sharing and storage of DNA/genomic data are often developed by research teams prioritising functionality delivered by the software rather than security of its operations. Consequently, such tools - that are processing such valuable data - are not developed as per the standards and best practises of safe and secure software.

Most open source tools that are being used in different research projects are written in unsafe and common languages like C and C++[1]. In 2017, researchers from the University of Washington evaluated different bioinformatics software tools in order to identify existing vulnerabilities [53]. The researchers investigated 13 tools from 6 different categories such as those for pre-processing, alignment, de novo assembly, alignment processing, RNA-seq and ChiP-seq. Their research highlighted that the programs written in C language contain a number of static buffers and also a huge number of insecure run-time library function calls thus making them highly vulnerable to buffer overflow attacks.

Tao et al [72] investigated the security vulnerabilities present in web-based bioinformatics applications. They concluded that most of these applications have versions that are outdated and are highly vulnerable to SQL injection attacks, XSS attack and file leakage, etc. One of their findings was that only 7.6%, making up one-fourth of entire websites they tested, are using secure HTTPS application-level protocol and the rest are using unsafe versions, such as HTTP which is concerning.

Existing vulnerabilities in software tools, websites and databases may result in a successful cyber-attack leading to

[1]The C programming language is widely regarded as 'unsafe' or not 'type safe' in the sense that it is not running in a environment where checks are performed during run-time. Flexibility in the C language to allow programmers to have low-level control over memory can result in many mistakes and program weaknesses not being identified during creation.

Keyword	Total Number	Exact Phrase	Exclude Citations	Exclude Patents	Exclude Citations and Patents
Need for Cyberbiosecurity	156	6	156	156	156
Security Privacy Cyberbiosecurity	76	0	76	79	79
Cyberbiosecurity of DNA Databases	41	0	41	41	41
Cyberbiosecurity and Genome Databases	59	0	56	57	57
Vulnerability Analysis Cyberbiosecurity	96	0	96	96	96
Cyberbiosecurity and DNA Databases	41	0	38	34	34
Security Analysis Bioinformatics Web Apps	2	2	2	2	2
Cyberbiosecurity Bioinformatics Tools	30	0	23	21	21
Need for Biocybersecurity	14	0	14	14	13
Information Assurance Genome	4	0	1	4	1
Computer Security DNA	2	0	2	2	2
Computer Security Genome	1	0	1	1	1

Table 1
Detailed literature review with keyword search

Online Databases	Contains Query or Service Tools	Include Analysis Pipeline Build	Access Control for Uploading Data	Access Control for Downloading Data	Requires Strong Password	Allows Programmatic Access
NCBI	Yes	No	Yes	No	No	Yes
EBI	Yes	No	Yes	No	No	Yes
PATRIC	Yes	Yes	Yes	No	No	Yes
DDBJ	Yes	No	Yes	No	No	Yes
EuPathDB	Yes	Yes	Yes	No	No	Yes
PAMDB	Yes	No	Yes	Yes	No	No

Table 2
Well-known online DNA and Genomic databases and their functions

the theft of an individual's genomic information that may cause consequences such as genetic discrimination or blackmail etc. While a stolen credit card can be replaced but an individual's genetic information cannot be replaces if stolen. Viantzer et al. [77] reviewed widely used genome databases, highlighting that although most of genomic databases require a username and password for access control; however, almost none of them require the user to select a strong password (long and including the use of capital letters, special characters, symbols and numbers). With weak passwords these systems are highly susceptible to a dictionary or brute force attack that can result in compromised passwords and stolen data. Another key finding by [77] was the use of MySQL queries, REST API and PERL API in some databases for remote users to directly query the data making these databases vulnerable to SQL-Injection attacks. Table 2 shows the summarized result of 2019 study [77] highlighting the vulnerabilities in widely known online genomic databases.

Edge et al. [21] mentioned two ways that an adversary can use to reveal genotypes information from genetic genealogy database. Firstly, an adversary can submit real genotype datasets, taken from publicly available databases, such as 1000 Genomes Project [22] or OpenSNP Project [56], to genealogical databases. This will result in bringing forth the data of genotype available in genealogy database that would match the data of publicly available genotype [41], thus compromising the confidentiality of private data uploaded on genetic genealogical databases. Alternatively, as highlighted by [53], an adversary can upload spoofed datasets to tamper these databases. These fake datasets are carefully designed by combining the chromosome information of two or more individuals from publicly available genetic catalogues to trick the underlying algorithm. This method can reveal the data of thousands of individuals using relative matching queries [53].

4.3. Scope of study

The primary focus of this research is on the bioinformatics tools and databases that are used in the bioinformatics pipeline. Figure 3 shows the schematic representation of steps to follow after collecting species sample and the role of bioinformatics tools and databases in a DNA lifecycle. After collecting hair, spit, blood sample etc. from the specie, DNA is extracted from that sample and is then sent to the lab for sequencing. At this stage, sequencing devices such as Illumina MiSeq is used which produces raw sequencing data which is then used in the bioinformatics pipeline. Bioinformatics pipeline contains software tools for analyzing, aligning, modeling etc. and it also contains database that can store genomic data.

An adversary can attack at multiple stages of the genomic data analysis pipeline as shown in Figure 3. From DNA sample collection to the usage of bioinformatics tools and databases each step contains vulnerabilities that could lead to the loss of data integrity or the DNA sample being completely wasted. Different types of attack are possible on each phase of the pipeline, such as physical attacks are possible on the sample collection and DNA extraction phase, hardware attacks are possible on the DNA sequencing phase, operating system attacks are possible on the raw data generation phase, and attacks on local or remote network are possible on the bioinformatic tools and databases. This scope of this study is to focus only on the vulnerabilities found in bioinformatics tools and databases and on the attacks those

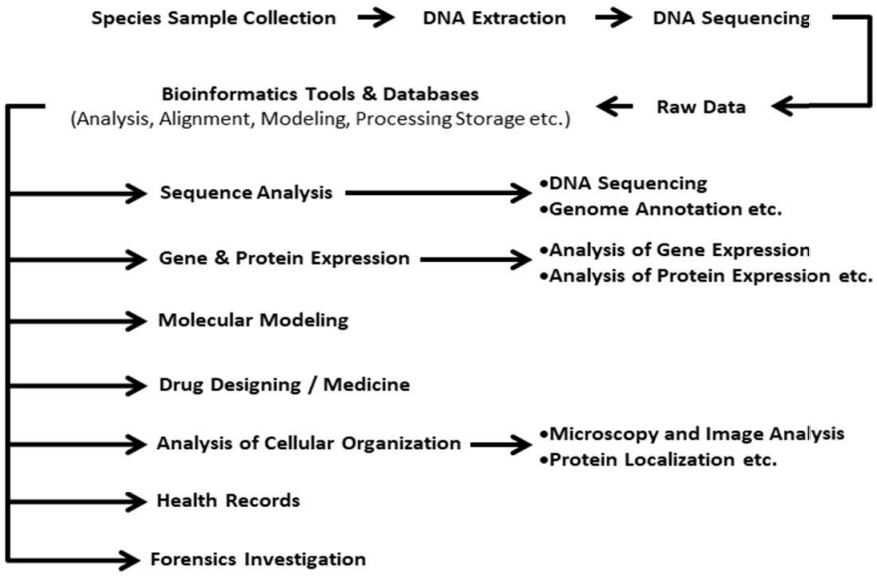

Figure 3: DNA sample lifecycle and the position of bioinformatics tools.

vulnerabilities could lead to.

Through an extensive and systematic review of literature presented above, we have concluded that there is a gap in the state of the art with respect to efforts focused at identifying and analysing vulnerabilities in genomic and DNA software. In particular, existing literature lacks a comprehensive taxonomy highlighting vulnerabilities in bioinformatics systems (data and software) and the possible cyber-attacks these vulnerabilities could lead to. Furthermore, existing efforts to highlight security vulnerabilities for bioinformatics software lack in-depth analysis supported by code analysis techniques such as static or dynamic code analysis which can help in not only identifying potential vulnerabilities but also to aid efforts evaluating likelihood of their exploitation.

This paper attempts to address this gap by conducting a thorough, scientific study into the security threats for genomic and DNA software to identify potential vulnerabilities (in the form of a taxonomy), to assess security hygiene of such software (through static code analysis), and to identify potential defence mechanisms to mitigate against threats identified.

5. Software security analysis of open-source bioinformatics tools and databases

In this section we evaluate the security of some widely used open source bioinformatics software's and databases. We have evaluated 25 different software that are widely used by biologists or bioinformaticians. The software we evaluated are all open source, written in C, C++, Java, Python and PHP and belong to different categories such as sequencing, alignment, alignment processing, pre-processing, and database. Our main focus was to identify the most common vulnerabilities that exist in these applications. We achieve this by performing static code analysis for bioinformatics chosen software to identify the security vulnerabilities. The list of tools used for static analysis is presented in Table 3.

As is evident from Table 4, our analysis result shows that most of the tools do not follow best practices for secure programming. For most of the software analysed, the code is written prioritizing functionality desired and issues such as performance optimization and security are not considered. The code written is not sanitized and poor coding practices such as improper placement of parenthesis, missing variable declaration etc. can be found in almost all the 25 tools we analyzed. Figure 4 shows some of the common programming errors found during analysis.

The most common vulnerability found in the tools written in C and C++ is buffer overflow. Use of static buffers and insecure library functions such as *strcat*, *strcpy*, *sprint*, *sprint*, *strlen*, *memcpy*, and *gets* etc. is common among these tools. We reviewed the tools as per OWASP recommendations for buffer overflow. Figure 5 shows the use of insecure library functions in a single class.

Command injections, Code injections and DoS attack vulnerabilities were found in bioinformatics software written in Python and JAVA. A tool that is commonly used as a Java library for bioinformatics also showed sign of race condition attack. A weak possibility of having a weak cryptography implementation can also be seen among these tools. Our analysis shows that databases and programs written in PHP show a clear sign of being vulnerable to SQL injection attack and XSS attack as shown in Figure 6.

Table 4 summarizes the vulnerabilities we found during static analysis of bioinformatics, DNA and genomic tools and databases. The table maps each tool with found common vulnerability present in them to the cyber attack that vulnerability could lead to. The focus of our study has been on open source software within bioinformatics and we have

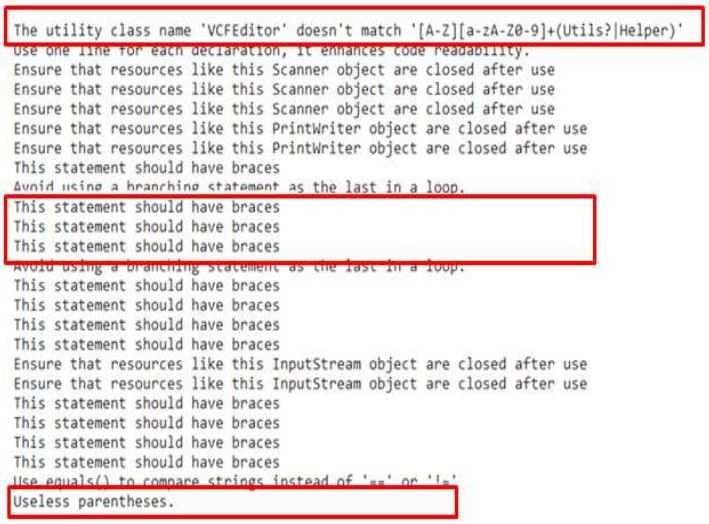

Figure 4: Code smells [26, 60] found in bioinformatics tools.

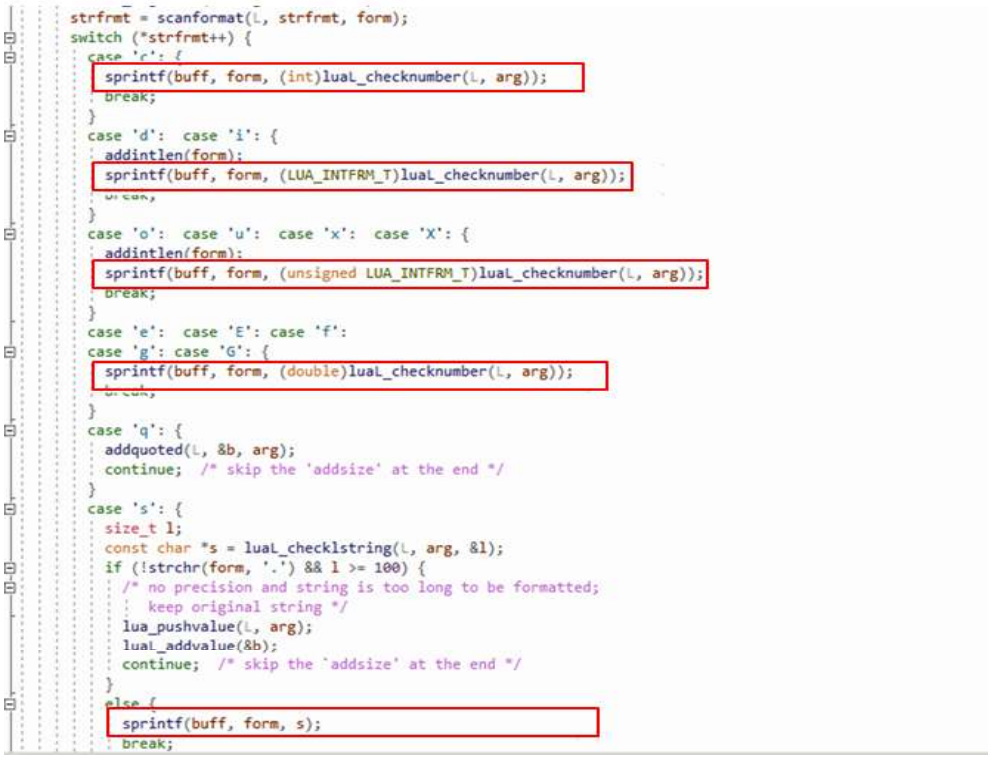

Figure 5: Use of unsafe C library functions in bioinformatics tools written in C.

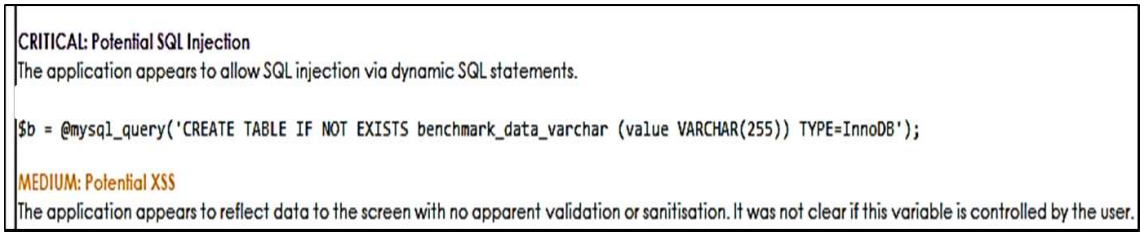

Figure 6: SQL injection and XSS vulnerability found in PHP tool.

S. No	Tool Name	Programming Language
1	Clang-query tool	C and C++
2	CppCheck	C and C++
3	VisualCodeGrepper	C, C++, PHP, Java etc.
4	CppDepend	C and C++
5	FindBugs	Java
6	Pylint	Python
7	Bandit	Python
8	SonarQube	Java, PHP, Python etc.
9	PMD (with security ruleset)	Java

Table 3
Software used for code analysis

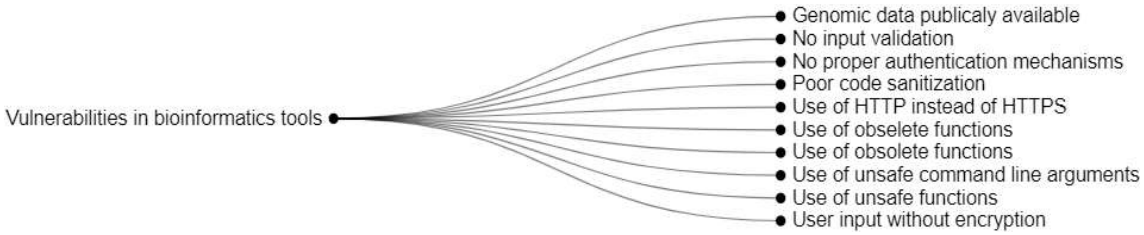

Figure 7: Taxonomy of vulnerabilities in bioinformatics tools

not been able to access proprietary software within this domain.

6. Proposed taxonomy of security and privacy issues in bioinformatics tools and databases

In this section, we propose the taxonomy of privacy and security related issues in bioinformatics tools and databases that process or store DNA and genomic data. This taxonomy is designed as a result of our analysis on open-source bioinformatics tools and databases and also by an extensive literature review of previous work done in this domain. Our taxonomy is broken into two parts, i.e. vulnerabilities and potential attacks. Figure 7 shows vulnerabilities that exist in these programs whereas Figure 8 shows the possible attacks that can exploit the vulnerabilities identified. The taxonomy consists of most common vulnerabilities found in these tools and databases and the common attacks they could result in.

6.1. Common Vulnerabilities:

1. **Poor code sanitization and lack of input validation:** Data sanitization and input validation may coexist and complement each other. Many online platforms processing or storing DNA and genomic data do not contain any mechanism to validate input or sanitize input or output. On the other hand, encoding of data is also necessary since DNA and genomic data is highly sensitive but it was also rarely implemented in the bioinformatics software analyzed.

2. **Use of unsafe, banned or obsolete functions:** Using unsafe, banned or obsolete functions such as *gets*, *scanf*, *strlen*, and *strcpy* etc. was one of the most common problems witnessed during the analysis of bioinformatics tools. The use of unsafe functions can lead to the security vulnerabilities, such as a buffer overflow. Once discovered, a vulnerability of this type could be exploited to cause adverse software behaviour.

3. **Inadequate authentication mechanism:** There is a lack of adequate authentication mechanisms implemented in many of the databases. Access controls have usually been implemented when uploading data but rarely applicable when downloading DNA and genomic data, which is considered personal and sensitive data [77]. This is significant as organisations have a legal obligation to take all reasonable steps to secure personal data, and neglecting to utilise stringent access control measures is negligent. Furthermore, when mechanisms such as passwords are used to secure access, it is important that both password policy and also implementation are robust. This includes reminding the user to choose a secure, yet memorable password [19]. In terms of implementation, the password should not be transferred or stored in plain-text form.

4. **Use of HTTP instead of HTTPS:** Some online platforms that contain DNA and genomic data do not use HTTPS. However, those who have implemented are are still vulnerable to cyber-attacks as identified by [72]. This is of great significance as data being transmitted using knowingly weak and vulnerable protocols is susceptible to attack, such as the man-in-the-middle attack, whereby data can be captured during plain-text transmission, i.e., it is not encrypted or secured.

5. **Genomic data publicly available:** Complete or partial DNA data, publicized DNA sequences, SNPS, phenotype or genotype are available through online databases or other online platform which can lead to chosen plain-text attacks. Furthermore, with the advancements in data analytics domain, it has become easier for malicious actors to collate different datasets to identify

Tools	Vulnerabilities	Attacks Possible
Genome Tools	Insufficient input validation / sanitation or unsafe user-supplied data concatenates into a SQL query	SQLI
	Insufficient input validation, No encoding	XSS
LOVD	Unsafe use of regular expressions (CVE-2017-16021 CVE-2018-13863)	Regular expression (Denial of service)
	Insufficient input validation / sanitation or unsafe user-supplied data concatenates into a SQL query	SQLI
	Insufficient input validation, No encoding	XSS
Crispor Website/ Crispor Paper	Unsafe use of regular expressions (CVE-2017-16021 CVE-2018-13863)	Regular Expression (Denial of Service)
	Unsafe command line arguments being passed to a system shell command (violating CVE-2018-7281, CVE-2018-12326, CVE-2011-3198)	Command injections
Predictd	Unsafe use of regular expressions (CVE-2017-16021 CVE-2018-13863)	Regular expression (Denial of service)
	Unsafe command line arguments being passed to a system shell command (violating CVE-2018-7281, CVE-2018-12326, CVE-2011-3198)	Command Injections
Starrpeaker	Unsafe command line arguments being passed to a system shell command (violating CVE-2018-7281, CVE-2018-12326, CVE-2011-3198)	Command injections
Glados	Unsafe use of regular expressions (CVE-2017-16021 CVE-2018-13863)	Regular expression (Denial of service)
	Unsafe command line arguments being passed to a system shell command (violating CVE-2018-7281, CVE-2018-12326, CVE-2011-3198)	Command injections
Online DNA and Genomic databases	Publicized DNA sequences, snps, phenotype or genotype available through online databases or any other online platform.	Correlation/ false relative attacks
		Identity tracing attacks
		Completion attacks
		Attribute disclosure attacks using DNA (ADAD)
		Inference attack
	No valid authentication mechanism	CSRF
	Using HTTP instead of HTTPS	Unauthorized Access
		DNS hijacking
		BGP hijacking
		Domain spoofing
Bioinformatics web applications	Using HTTP instead of HTTPS	Unauthorized access
		DNS hijacking
		BGP hijacking
		Domain spoofing
BioJava	The application appears to create a temporary file with a static, hard-coded name.	Race condition attack
	Unsafe use of regular expressions (CVE-2017-16021 CVE-2018-13863)	Regular expression (Denial of service)
Hoffman Lab	Use of Unsafe or banned Functions	Buffer overflow Attack
HapCut2		
Freebayes		
BitSeq		
Bamtools		
Sniffles		
Survivor		
NGLMR		
Minimap		
Bowtie2		
Bowtie		
Samtools		
Bwa		
Fqzcomp		
STAR		
BLAT		

Table 4
Vulnerabilities identified within DNA-genomics tools

hidden patterns in the data which may reveal personally identifiable information.

6. **Use of unsafe command line arguments:** Unnecessary or unsafe command line arguments were also found to be used in many tools.

6.2. Potential Cyber-attacks:

1. **Denial of Service attacks:** Denial of Service (DoS) is an attack to deny a victim access to a particular resource or service, and has become one of the major threats and rated among the most challenging internet security issues [20]. Some of the relevant DoS attacks for bioinformatics software are highlighted below:

(a) **Application layer attacks:** Common application layer attacks include BGP hijacking, low and slow attack, GET/POST floods etc. They are low-volume attacks aiming to crash the web server through sending too many requests.

(b) **Protocol attacks:** Common protocol attacks include SYN floods, Ping of Death, Smurf DDoS etc. This type of attack consumes server resources to the point where it can no longer respond to legitimate users.

(c) **Volume based attacks:** Common volume based attacks include UDP floods, ICMP floods etc. Its goal is to saturate the bandwidth of the attacked site.

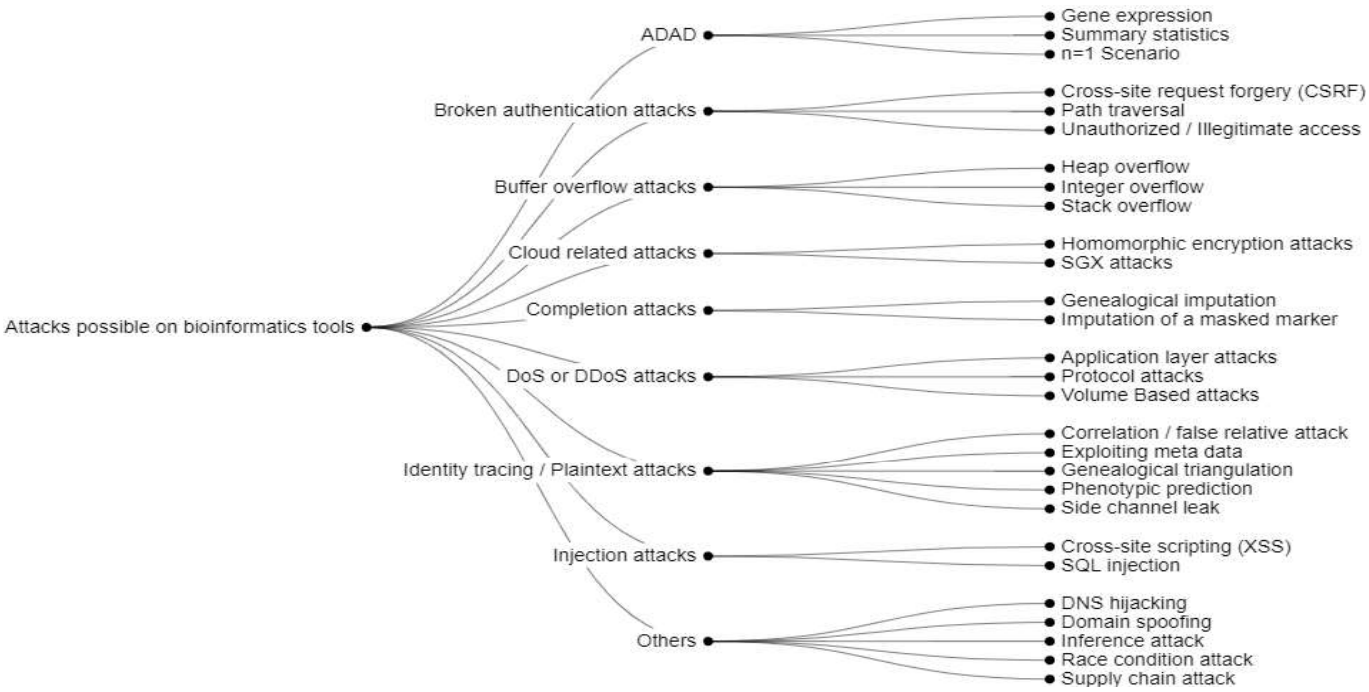

Figure 8: Taxonomy of attacks in bioinformatics tools

2. **Broken authentication attacks:** An attacker can take advantage of an insufficient authentication validation vulnerability and capture user credentials to impersonate a valid user, which is one of the common cyber-attacks affecting web applications. Some of the potential web-based attacks for bioinformatics software are described below:
   (a) **Cross site request forgery (CSRF):** A very common web application attack that forces end users to execute undesired actions. A successful CSRF attack can force a user to perform state changing requests like transferring funds, changing their email address, etc. If the victim is an administrative account, CSRF can compromise the entire web application [58].
   (b) **Path traversal:** Such attacks try to access files that are normally not directly accessible by anyone i.e. try to access confidential data. Such attacks can be submitted in URLs, system calls or in shell commands.
   (c) **Unauthorized illegitimate access:** Authentication flaw within an application allowing the compromise of user related credentials to portray a valid user's identity (ie. Keys, passwords, tokens, etc.) [67].
3. **Attribute disclosure attacks using DNA (ADAD):** ADAD attacks use genetic markers or characteristics to identify individuals and disclose further information about them. Potential ADAD attacks for bioinformatics software are explained below.
   (a) **Gene expression:** A type of ADAD attack, the essence of Gene expression attacks is to learn loci in gene expression profiles that are the most probable markers of a given genotype. Once such markers are learned, they can be used to compare anonymised gene expression datasets to medical data with patient information.
   (b) **Summary statistics:** Homer et al. [36] highlighted the possibility of ADAD on Genome-wide association studies (GWAS) data sets that only consist of the allele frequencies of the study participants. The underlying concept of summary statistics scenario for ADAD can be understood by considering an extremely rare variation in the subject's genome – a non-zero allele frequency of this variation in a small study increases the likelihood that the target was part of the study, whereas a zero allele frequency strongly reduces this likelihood.
   (c) **n=1 scenario:** n=1 scenario in ADAD is one where the sensitive attribute in the dataset is associated with the genotype data of the individual. In this case, the adversary can simply match the genotype data that is associated with the identity of the individual and that is associated with the attribute. Genome-wide association studies (GWAS) are highly vulnerable to such attacks [25].
4. **Buffer overflow attack:** A buffer overflow occurs when the buffer contains more data than it can handle resulting in data overflow into adjacent storage. It is the

most common form of security vulnerability found in software programs [18]. Buffer overflows can be exploited by attackers to corrupt software. In a buffer overflow attack, a malicious user exploits a program buffer with weak or no bounds checking and overwrites the program code with their own data or executable code hence changing the programs operation for their gain. This change can cause system to crash or can create an entry point for a cyber attack. It can occur in both stack and heap memory locations.

5. **Completion attacks:** Completion of genetic information from partial data is a common problem. To create the missing genotypic values a combination of linkage disequilibrium between markers and reference panels with complete genetic information can be used. This can be misused by adversaries.

6. **Identity tracing attacks:** Identity tracing attacks attempt to uniquely identify an anonymous DNA sample using quasi-identifiers from the dataset [25]. Potential identity tracing attacks for bioinformatics software are explained below.

   (a) **Correlation attacks:** Using 3rd party DNA testing services can result in an individuals DNA data being stolen from database or used without consent. Researchers from Washington university [52] demonstrated false relative attack by using a small number of specifically designed files and extracted genetic markers including medically sensitive markers from other users. They were able to successfully extract 92% of markers and then compared extracted and target profiles manually to acquire a person's entire profile. Attackers can do the same by creating a false relative sample that falsely mimics the relative of existing sample and steal sensitive data.

   (b) **Exploiting meta data:** Genomic datasets are often published with additional metadata, which can be exploited to trace the identity of an unknown genome in the sample. Demographic metadata is a strong source of identifying information. According to a past study [70], it was estimated that the combination of gender, date of birth and zip code is enough to uniquely identify 60% of the US nationals.

   (c) **Genealogical triangulation:** With the development of online platforms and databases to search for genetic matches genealogical triangulation has become a viable attack for identity tracers. Surnames are passed from father to son in most of the societies, this creates a transient correlation with specific Y chromosome called haplotypes [25]. Attackers can take advantage of the Y chromosome surname correlation and compare the Y-chromosome haplotype of the unknown genome to haplotype records in genetic genealogy databases. An example of such attack was demonstrated in the year 2013 by scientists [32]. They performed surname-inference attack to exploit the Y chromosome surname correlation.

   (d) **Phenotypic prediction:** It is envisioned that the prediction of phenotypes from genetic data could be used as quasi identifiers for tracing.

   (e) **Side channel leaks:** This attack exploits unintentionally coded quasi identifiers in datasets, instead of targeting the actual data that is made public. These attacks exploit factors such as filenames, numbering, hash values, and other basic computer security vulnerabilities, to discover further information about participants in a genomic dataset. In 2013 scientist reported that the uncompressed files from the Personal Genome Project (PGP) have filenames that contain the actual name of the study participant [71], making it a strong target for attackers.

7. **Injection attacks:** A program fails to validate the input sent to the program from a user. An attacker can exploit an insufficient input validation vulnerability and inject arbitrary code, which commonly occurs within web applications. Potential injection attacks for bioinformatics software are explained below.

   (a) **SQL injection attack:** SQL injection vulnerabilities have been labeled as one of the most severe threats for web applications or websites [57]. Through SQL injection attacks, attackers can execute malicious SQL statements to obtain unrestricted access of databases containing sensitive data resulting in security violations in the form of identity theft, loss of information and fraud [33].

   (b) **Cross site scripting attack (XSS):** XSS flaws involve a design flaw not properly validated allowing malicious scripts to be executed against a vulnerable application in a web browser.

8. **Cloud related attacks:** Keeping in view of the huge amount of genomic data and the requirement of heavy processing capabilities there have been a number of cloud based solutions evolved for facilitating genomic reserachers. For example, DNAnexus [65], Galaxy [1], and Bionimbus [35] are among such genomic cloud computing platforms. However, there have been numerous security risks pertaining to the utilisation of genomic cloud services because of the highly sensitive nature of genomic data. In a cloud environment users are always concerned about the security and privacy of their data which resides beyond their physical boundaries. The unauthorised access of data for the purpose of reuse or misuse, the modification or corruption of data, failure of accountability, and unavailability of data are among the most serious risks in the cloud environment.

(a) **SGX attacks:** The Intel Software Guard Extension (SGX) is one of the latest innovative methods for protecting sensitive data in a distributed environment. Recently there have been a number of successful attacks demonstrated against the security of SGX. For example, the side-channel attacks are considered practically achievable by the malicious softwares that reside on the same machine where the SGX enclaves are created [17] [28] [48]. Keeping in view of the constantly evolving attack space a more rigorous security evaluation of SGX is required.

(b) **Homomorphics encryption attacks:** Homomorphic encryption has recently gained a lot of attention because of its capability of protecting sensitive data in a distributed environment. It can be employed to protect confidentiality of data during its processing by a cloud service [13] [76] [29]. However, besides these major contributions attacks are still reported against the security of homomorphic encryption as recently presented by Baiyu Li and Daniele Micciancio in [43].

9. **Software supply chain attacks:**
A software supply chain attack is characterised by the hacker's action to inject malicious code in software components to infect the whole downstream software chain. Recently, an important software supply chain security breach happened after a successful hacking of SolarWinds Orion platform in March 2020 that is widely used infrastructure monitoring and management platform in United States. The security breach was discovered by a cybersecurity firm FireEye after a few months of actual incident. The attack was allegedly launched by the hackers group named 'APT29' or 'Cozy Bear' through a trojan horse into the software update server of SolarWinds software supply chain system [5]. Consequently, a malware infected a large number of computers used by highly sensitive departments and agencies of U.S. government. Such software supply chain security breaches have become extremely dangerous because of the increasing adoption of open source software packages. There have been numerous malicious software packages that have reportedly been used to infect the open source software supply chains through well-known repositories such as RubyGems and PyPI [55]. Hence, it is extremely important for the software developers to carefully use version pinning and avoid as much as possible from automated software updates and bug fixes to avoid such security breaches. Similarly, proper auditing of unapproved IT assets, maintenance of updated reliable software inventory, assessment of vendor's security posture and adoption of client side protection tools such as RASP. There have been a number of recent research contributions on the security of software supply chain in medical and healthcare sector [78]. Few of such examples include the cross-site genomic data access using blockchain technology [44] [27], decentralised genomic data sharing [66] and secure management of genomic data using blockchain [37]

10. **Others:**

(a) **DNS hijacking:** Type of DNS attack in which attacker redirects queries to a different domain name server. It targets the DNS record of the website on the nameserver. This can be done with malware or with the unauthorized modification of a DNS server.

(b) **Domain spoofing:** A type of phishing attack with the goal of stealing personal information, to trick the victim into sending money to the attacker, or to trick a user into downloading malware.

(c) **Inference attack:** Inference attack can compromise kin genomic privacy [7]. Adversary is assumed to launch inference attacks with extensive available knowledge of the known SNPs from individuals who share their SNPs,the known traits shared by individuals, the GWAS catalog which contains the interdependent information among traits and SNPs and statistical information.

(d) **Race condition attack:** Occurs when a program or system that's designed to handle tasks in a prespecified sequence is forced to perform two or more operations simultaneously. This technique takes advantage of a time gap between the moment a service is initiated and the moment a security control takes effect.

7. Methods and practices to protect DNA and genomic data

Genomic data of an individual contains sensitive and private information. It can be used to track an individual's ancestors or relatives and also contains information about possible genetic diseases making it sensitive personal information. In this section, we share existing methods, techniques and practices that can be implemented to protect the confidentiality and maintain the integrity of genomic data from cyber-attacks. Leveraging the work presented in section 5 and 4.2, we present a mapping between vulnerabilities, attacks and defence mechanisms in Table 5 and explain potential defence mechanisms below.

1. **Laws and frameworks:** The genomic data is recognised as personally identifiable sensitive information across different legal regulations and frameworks. With respect to security and privacy of genomic data, most commonly used legal frameworks include Health Insurance Portability and Accountability Act (HIPPA) and Genetic Information Nondiscrimination Act (GINA).

Vulnerabilities	Attack Classification	Attack Types	Defense Mechanism
Poor code sanitization /user input without validation or encryption	Injection attacks	SQL injection attacks	Parameterize queries Encode data Validate all inputs Implement logging Intrusion detection Always log the timestamp
		XSS - Cross site scripting	
	Broken authentication attacks	CSRF-Cross site request forgery	
Use of unsafe functions	Buffer overflow attack	Stack overflow attack	Structured exception handler overwrite protection Data execution prevention Address space randomization Code in languages with built-in buffer overflow protection
		Heap overflow attack	
		Integer overflow attack	
	DoS or DDoS attack	Volume based attacks	Activate a website application firewall protection Country blocking Monitor traffic Block application layer DDoS attacks
		Protocol attacks	
		Application layer attacks	
	Other	Supply chain attack	Implement strong code integrity policies and use endpoint detection.
Improper or no authentication	Broken authentication	Unauthorized access	Encode data Validate all inputs Implement logging Implement intrusion detection Implement identity and authentication controls Implement HTTPS (SSL/TLS) Use framework that supports server-side trusted data for driving access control Use biometric authentication Use secure password storage Password hashing Implement secure password recovery mechanism Establish timeout / inactivity periods Use re-authentication for sensitive features Use monitoring and analytics to spot suspicious IPs and machine IDs
		CSRF- Cross site request forgery	
		Path traversal	
	Cloud related attacks	SGX attacks	
		Homomorphic encryption attacks	
	Other	Race condition attack	
Use of HTTP instead of HTTPs	Other	DNS hijacking	Adopt SSL/TLS Use flow telemetry analysis supplemented with behavioral analysis to detect abnormalities Use an IDMS to detect abnormal behavior
	Application-layer DDoS attacks	BGP hijacking	
	Other	Domain spoofing	
Genomic data publicly available	Other	Inference attack	Use re-authentication for sensitive or highly secure features Implement identity and authentication controls Establish timeout and inactivity periods for every session Implementation of policy for publicly available data Laws and frameworks should be enforced to protect data
	Identity tracing / Plaintext attacks	Exploiting meta data	
		Genealogical triangulation	
		Phenotypic prediction	
		Side channel leaks	
		Correlation / false relative attack	
	Completion attacks	Genealogical imputation	
		Imputation of a masked marker	
	Attribute disclosure attacks using DNA	Gene expression	
		Summary statistics	
		n=1 scenario	
Use of obsolete functions	Other	Information disclosure (Loss of private data)	Actively review or maintain code to remove obsolete functions Software testing should be done before releasing it for public use Avoid use of unsafe command line arguments
Use of unsafe command line arguments			

Table 5
Mapping of vulnerabilities to attack types and defense mechanisms

For instance, GINA is focused at protecting individuals from genomic discrimination that often comes from the employers and health insurers. Although these frameworks recognise the significance of genomic data and the need to protect such data against cyber-threats, adoption of thes frameworks within security policies of organizations dealing with genomic data is crucial.

2. **Genomic data governance:** The increase in use of cutting-edge technologies has also introduced signif-

icant challenges such as legal, ethical and privacy violations and misuse to name a few. In this context, there is a need for adequate governance framework to be able to mitigate the trade-off between ease of sharing and processing and security and privacy of the genomic data. As highlighted by Kieran et al [59], notable genomic data projects adopt custom governance structures which are focused at characteristics of the individual projects. However, these governance frameworks lack adequate attention to cyber-threats (internal and external threat actors). Furthermore, although Porgram for Engaging Everyone Responsibly (PEER) [62] leverages GDPR and CCPA, there is a lack of standardisation across various genomic data initiatives.

3. **Encryption / cryptographic solutions:** One of the major concerns identified through our analysis is the insecure data storage and sharing. In recent years, a number of efforts such as [11, 9] have focused on developing privacy enhancing techniques for genomic data. However, a defence in depth approach is highly desirable, taking a holistic view of the threat landscape for genomic data processing. Therein, with the increase in adoption of web-based technologies to achieve efficient processing of genomic data, use of cryptographic techniques to secure data at rest and in motion is significant. Our analysis identified that not all bioinformatics tools use cryptographic mechanisms to secure data in motion. Furthermore, those using HTTPS for secure communication may be compromising on the strength of cryptographic mechanisms to enhance usability. In view of the significance of the genomic data, the use of strong cryptographic measures should be encouraged and adopted as standard within this domain.
4. **Identification & authentication controls:** Identification and authentication form the first line of defence typically for web-based applications. However, due to the complexity in usability of strong passwords mechanisms such as multi-factor authentication and biometrics have been increasingly used to achieve authentication mechanisms resilient against cyber-attacks. Our analysis has particularly highlighted the need for advanced authentication controls within bioinformatics software to achieve enhanced resilience against cyber-attacks. Unlike with some systems, access is being restricted to highly sensitive and personal data, therefore it is critical to ensure that a robust password policy is implemented and the system uses hashing techniques to ensure passwords are not transferred in plain-text [61]. Hashing algorithms such as PBKDF2 are widely used and are regarded to have desirable security properties and low computation requirements [34].
5. **Access controls:** The implementation of access control schemes and policies within the health care organizations that hosts DNA and genomic databases could greatly improve genomic privacy [25]. National Center for Biotechnology Information (NCBI) hosts genotypes and phenotypes databases in order to protect these databases NCBI has an access control policy in place which states that violating these conditions would result in access being revoked, as well as other potential penalties. Our analysis of bioinformatics software revealed lack of appropriate access control measures for most of those analysed therefore highlighting avenues of further work.
6. **Input sanitation and validation:** Our analysis of bioinformatics software has highlighted lack of input validation (within web-based applications) as one of the core vulnerabilities which may lead to code injection attacks. Therefore, input sanitation and validation mechanisms should be adopted to protect against such attacks. In this regard, *whitelisting* and *blacklisting* are two commonly used approaches. Blacklisting attempts to check that given data does not contain undesired content i.e. sanitation. However, whitelists can be developed to perform specific checks on input validation. Example of whitelisting is the use of regular expressions as they can check whether data matches pre-defined pattern that is required for your system. Additionally, client and server-side input validation is also needed taking into account HTTP headers, file uploads, GET/POST parameters, and cookies etc.
7. **Query parameterization:** Query parameterization is a technique that can be used to avoid SQL injection attacks. Our analysis of bioinformatics tools indicated the lack of prepared statements or parameterized queries.
8. **Intrusion detection and logging:** Intrusion detection and monitoring systems form important components of a security architecture and provide defence-in-depth. Such systems are focused at recording system events and identifying patterns of misuse. A major benefit of such mechanisms is that by recording system events, they provide an opportunity to learn from new ways of system use and adapt accordingly thereby facilitating protection against zero-day attacks.
9. **Buffer overflow protection and code hygiene:** A major security concern we identified through our analysis of bioinformatics software is their susceptibility to buffer overflow attacks. Therefore, appropriate protection mechanisms are required to mitigate against such attacks. In this context, mechanisms such as *structured exception handler overwrite protection (SEHOP)*, *address space randomization (ASR)*, and *data execution prevention* can be used. Furthermore, our analysis also highlighted that some bioinformatics software were using deprecated or obsolete functions. This indicates that the code has not been actively reviewed or maintained and therefore requires removal of deprecated functions and regular software testing to identify such occurrences.
10. **Informed consent:** With the advancements in access to and processing of genomic data, discussion around measures to safeguard such data has intensified. In particular, as it is considered personal data, the chal-

lenge of privacy is particularly significant. Therefore, researchers must find a way to publish genetic data in a way that it maintains individuals' privacy but still has scientific value. Those who publish their genomic information, or participate in such studies, should be made aware of the implications i.e. appropriate mechanisms to manage consent should be adopted. In this respect, informed consent applies to: i) an individual's decision to disclose personal data, ii) decisions about controls on access to the data, and iii) decisions about appropriate uses (and what constitutes "misuse") of the data.

11. **Data-in-use protection:**
Introduced in 2015 the Intel Software Guard Extension (SGX) [17] is an innovative solution for achieving privacy and security of sensitive data at the processor level. SGX provides a set of instruction codes for applying security (in terms of confidentiality and integrity) on highly sensitive data at runtime by creating separate and private memory regions called *enclaves*. These enclaves provide isolated environment to prevent unauthorised reading and storing of secret memory content from untrusted operating system during the execution time. With SGX the operating system cannot have a direct access to the enclave because it is only decrypted at runtime within the CPU. Another emerging trend in privacy and security of data is homomorphic encryption which is being widely adopted due to its enormous advantages [4] [3]. Homomorphic encryption enables processing of encrypted data without compromising its confidentiality which is ideal for processing data in a distributed computing environment. An important contribution is the use of homomorphic encryption for the privacy of genomic data that is processed over the cloud in an untrusted environment [63] [38]. The authors presented the privacy preserving computation on encrypted confidential DNA sequences and compared the performance of different approaches of practical homomorphic encryption methods.

8. Conclusion and future work

In this paper we have highlighted the need for cyber-biosecurity. We have explored the common vulnerabilities found in bioinformatics, DNA and genomic tools and databases by performing static analysis on 25 open source tools written in C, C++, Java etc. and have mapped those vulnerabilities to the attacks they could lead to. We also, for the first time, proposed a taxonomy of security and privacy issues in bioinformatics tools and databases. Our work also suggests methods and practices that can be implemented to eliminate found weaknesses.

Due to the fact that the stolen DNA and genomic data can have severe effects, the need to protect the applications and databases processing, storing or synthesizing DNA and genomic data is a must. Vulnerabilities need to be addressed, more work needs to be done in this field. After static analysis, dynamic analysis can be performed to further investigate existing flaws in these applications. Proposed taxonomy can be expanded after in depth dynamic analysis is done. Furthermore, likelihood of attacks is an important factor which can contribute to the security of the genomic data analysis pipeline. This paper has not included this as likelihood of an attack is highly subjective and depends upon a number of factors including the overall security posture of the victim/target, the resources (time, money, and computation etc) available to the attacker, and the skill aptitude of the attacker. A comprehensive solution to this complex challenge will require an in-depth analysis of these factors to arrive at a reliable scoring system. Therefore, we consider this an opportunity for future work.

References

[1] Afgan, E., Baker, D., Coraor, N., Goto, H., Paul, I.M., Makova, K.D., Nekrutenko, A., Taylor, J., 2011. Harnessing cloud computing with galaxy cloud. Nature biotechnology 29, 972–974.

[2] Aiba, Y., Sumaoka, J., Komiyama, M., 2011. Artificial dna cutters for dna manipulation and genome engineering. Chemical Society Reviews 40, 5657–5668.

[3] Alaya, B., Laouamer, L., Msilini, N., 2020. Homomorphic encryption systems statement: Trends and challenges. Computer Science Review 36, 100235.

[4] Alloghani, M., Alani, M.M., Al-Jumeily, D., Baker, T., Mustafina, J., Hussain, A., Aljaaf, A.J., 2019. A systematic review on the status and progress of homomorphic encryption technologies. Journal of Information Security and Applications 48, 102362.

[5] Analytica, O., . Solarwinds hack will alter us cyber strategy. Emerald Expert Briefings .

[6] Ashcroft, R., 2007. Should genetic information be disclosed to insurers? no. BMJ 334, 1197–1197.

[7] Ayday, E., Humbert, M., 2017. Inference attacks against kin genomic privacy. IEEE Security and Privacy 15, 29–37.

[8] Benson, D.A., Cavanaugh, M., Clark, K., Karsch-Mizrachi, I., Ostell, J., Pruitt, K.D., Sayers, E.W., 2018. Genbank. Nucleic acids research 46, D41–D47.

[9] Berger, B., Cho, H., 2019. Emerging technologies towards enhancing privacy in genomic data sharing.

[10] Blog, A.M., . Myheritage statement about a cybersecurity incident. https://blog.myheritage.com/2018/06/myheritage-statement-about-a-cybersecurity-incident/.

[11] Bonomi, L., Huang, Y., Ohno-Machado, L., 2020. Privacy challenges and research opportunities for genomic data sharing. Nature genetics 52, 646–654.

[12] Buiten, M.C., 2019. 'your dna is one click away': The gdpr and direct-to-consumer genetic testing, in: Consumer Law and Economics. Springer, pp. 205–223.

[13] Chatterjee, A., Aung, K.M.M., 2019. Translating algorithms to handle fully homomorphic encrypted data, in: Fully Homomorphic Encryption in Real World Applications. Springer, pp. 49–70.

[14] Christofides, E., O'Doherty, K., 2016. Company disclosure and consumer perceptions of the privacy implications of direct-to-consumer genetic testing. New Genetics and Society 35, 101–123.

[15] Clayton, E.W., Evans, B.J., Hazel, J.W., Rothstein, M.A., 2019. The law of genetic privacy: applications, implications, and limitations. Journal of Law and the Biosciences 6, 1–36.

[16] Colin Mitchell, Johan Ordish, E.J.T.B., Hall, A., 2020. The gdpr and genomic data.

[17] Costan, V., Devadas, S., 2016. Intel sgx explained. IACR Cryptol. ePrint Arch. 2016, 1–118.

[18] Cowan, C., Wagle, F., Pu, C., Beattie, S., Walpole, J., 2000. Buffer

overflows: Attacks and defenses for the vulnerability of the decade, in: Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00, IEEE. pp. 119–129.
[19] Dell'Amico, M., Michiardi, P., Roudier, Y., 2010. Password strength: An empirical analysis, in: 2010 Proceedings IEEE INFOCOM, IEEE. pp. 1–9.
[20] Douligeris, C., Mitrokotsa, A., 2004. Ddos attacks and defense mechanisms: classification and state-of-the-art. Computer Networks 44, 643–666.
[21] Edge, M.D., Coop, G., 2020. Attacks on genetic privacy via uploads to genealogical databases. Elife 9.
[22] EMBL-EBI, (accessed December 5, 2019). Igsr and the 1000 genomes project. https://www.internationalgenome.org/.
[23] Emily Darraj, B.M., 2017(accessed December 1, 2019). Genomic data requires better protection. http://health21initiative.org/article/genomic-data-requires-better-protection.
[24] Ensembl, (accessed February 28, 2020). Genome browser. https://asia.ensembl.org/index.html.
[25] Erlich, Y., Narayanan, A., 2014. Routes for breaching and protecting genetic privacy. Nature Reviews Genetics 15, 409–421.
[26] Fowler, M., 2018. Refactoring: improving the design of existing code. Addison-Wesley Professional.
[27] Gammon, K., 2018. Experimenting with blockchain: can one technology boost both data integrity and patients' pocketbooks?
[28] Ge, Q., Yarom, Y., Cock, D., Heiser, G., 2018. A survey of microarchitectural timing attacks and countermeasures on contemporary hardware. Journal of Cryptographic Engineering 8, 1–27.
[29] Geng, Y., et al., 2019. Homomorphic encryption technology for cloud computing. Procedia Computer Science 154, 73–83.
[30] GenomicsEngland, (accessed March 24, 2020). The 100,000 genomes project. https://www.genomicsengland.co.uk/about-genomics-england/the-100000-genomes-project/.
[31] Guttmacher, A.E., Collins, F.S., 2003 (accessed March 01,2020). Welcome to the genomic era. https://www.nejm.org/doi/full/10.1056/NEJMe038132.
[32] Gymrek, M., McGuire, A.L., Golan, D., Halperin, E., Erlich, Y., 2013. Identifying personal genomes by surname inference. Science 339, 321–324.
[33] Halfond, W.G., Viegas, J., Orso, A., et al., 2006. A classification of sql-injection attacks and countermeasures, in: Proceedings of the IEEE international symposium on secure software engineering, IEEE. pp. 13–15.
[34] Hatzivasilis, G., 2017. Password-hashing status. Cryptography 1, 10.
[35] Heath, A.P., Greenway, M., Powell, R., Spring, J., Suarez, R., Hanley, D., Bandlamudi, C., McNerney, M.E., White, K.P., Grossman, R.L., 2014. Bionimbus: a cloud for managing, analyzing and sharing large genomics datasets. Journal of the American Medical Informatics Association 21, 969–975.
[36] Homer, N., Szelinger, S., Redman, M., Duggan, D., Tembe, W., Muehling, J., Pearson, J.V., Stephan, D.A., Nelson, S.F., Craig, D.W., 2008. Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. PLoS Genet 4, e1000167.
[37] Jin, X.L., Zhang, M., Zhou, Z., Yu, X., 2019. Application of a blockchain platform to manage and secure personal genomic data: a case study of lifecode. ai in china. Journal of medical Internet research 21, e13587.
[38] Kim, M., Lauter, K., 2015. Private genome analysis through homomorphic encryption, in: BMC medical informatics and decision making, BioMed Central. pp. 1–12.
[39] Kornilov, S.A., Tan, M., Aljughaiman, A., Naumova, O.Y., Grigorenko, E.L., 2019. Genome-wide homozygosity mapping reveals genes associated with cognitive ability in children from saudi arabia. Frontiers in Genetics 10, 888.
[40] Kruse, C.S., Frederick, B., Jacobson, T., Monticone, D.K., 2017. Cybersecurity in healthcare: A systematic review of modern threats and trends. Technology and Health Care 25, 1–10.
[41] Larkin, L., 2017. Cystic fibrosis: A case study in genetic privacy. The DNA Geek .
[42] Ledford, H., 2015. Crispr, the disruptor. Nature News 522, 20.
[43] Li, B., Micciancio, D., 2020. On the security of homomorphic encryption on approximate numbers. IACR Cryptol. ePrint Arch 2020, 1533.
[44] Ma, S., Cao, Y., Xiong, L., 2020. Efficient logging and querying for blockchain-based cross-site genomic dataset access audit. BMC Medical Genomics 13, 1–13.
[45] Malin, B., Benitez, K., Masys, D., 2011. Never too old for anonymity: a statistical standard for demographic data sharing via the hipaa privacy rule. Journal of the American Medical Informatics Association 18, 3–10.
[46] of Medicine, N.L., (accessed January 7, 2020). What is genome? https://ghr.nlm.nih.gov/primer/hgp/genome.
[47] Meller, R., 2015. Addressing benefits, risks and consent in next generation sequencing studies. Journal of clinical research and bioethics 6.
[48] Moghimi, A., Wichelmann, J., Eisenbarth, T., Sunar, B., 2019. Memjam: A false dependency attack against constant-time crypto implementations. International Journal of Parallel Programming 47, 538–570.
[49] Murch, R.S., So, W.K., Buchholz, W.G., Raman, S., Peccoud, J., 2018. Cyberbiosecurity: an emerging new discipline to help safeguard the bioeconomy. Frontiers in bioengineering and biotechnology 6, 39.
[50] NCBI, (accessed February 28, 2020). Genbank. https://www.ncbi.nlm.nih.gov/genbank/.
[51] News, H.I., 2020 (accessed July 25, 2020). Ransomware: See the 14 hospitals attacked so far in 2016. https://www.healthcareitnews.com/slideshow/ransomware-see-hospitals-hit-2016.
[52] Ney, P., Ceze, L., Kohno, T., 2020. Genotype extraction and false relative attacks: security risks to third-party genetic genealogy services beyond identity inference, in: Network and Distributed System Security Symposium (NDSS).
[53] Ney, P., Koscher, K., Organick, L., Ceze, L., Kohno, T., 2017. Computer security, privacy, and {DNA} sequencing: Compromising computers with synthesized {DNA}, privacy leaks, and more, in: 26th {USENIX} Security Symposium ({USENIX} Security 17), pp. 765–779.
[54] NIH, (accessed March 24,2020). https://www.nih.gov/.
[55] Ohm, M., Plate, H., Sykosch, A., Meier, M., 2020. Backstabber's knife collection: A review of open source software supply chain attacks, in: International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, Springer. pp. 23–43.
[56] OpenSNP, (accessed December 5, 2019). Opensnp project. https://opensnp.org/.
[57] OWASP, 2020 (accessed July 15, 2020). Top 10 web application security risks. https://owasp.org/www-project-top-ten/.
[58] OWASP, 2020 (accessed June 02, 2020). Cross site request forgery (csrf). https://owasp.org/www-community/attacks/csrf.
[59] O'Doherty, K.C., Shabani, M., Dove, E.S., Bentzen, H.B., Borry, P., Burgess, M.M., Chalmers, D., De Vries, J., Eckstein, L., Fullerton, S.M., et al., 2021. Toward better governance of human genomic data. Nature Genetics 53, 2–8.
[60] Paiva, T., Damasceno, A., Figueiredo, E., Sant'Anna, C., 2017. On the evaluation of code smells and detection tools. Journal of Software Engineering Research and Development 5, 7.
[61] Peyravian, M., Zunic, N., 2000. Methods for protecting password transmission. Computers & Security 19, 466–469.
[62] Project, P., 2021 (accessed March 09, 2021). Promise for engaging everyone responsibly. http://geneticalliance.org/programs/biotrust/peer.
[63] Raisaro, J.L., Choi, G., Pradervand, S., Colsenet, R., Jacquemont, N., Rosat, N., Mooser, V., Hubaux, J.P., 2018. Protecting privacy and security of genomic data in i2b2 with homomorphic encryption and differential privacy. IEEE/ACM transactions on computational biology and bioinformatics 15, 1413–1426.
[64] Regalado, A., 2019 (accessed March 20, 2020). Mit technol-

ogy review "more than 26 million people have taken an at-home ancestry test". https://www.technologyreview.com/s/612880/more-than-26-million-people-have-taken-an-at-home-ancestry-test/.

[65] Reid, J.G., Carroll, A., Veeraraghavan, N., Dahdouli, M., Sundquist, A., English, A., Bainbridge, M., White, S., Salerno, W., Buhay, C., et al., 2014. Launching genomics into the cloud: deployment of mercury, a next generation sequence analysis pipeline. BMC bioinformatics 15, 1–11.

[66] Shabani, M., 2019. Blockchain-based platforms for genomic data sharing: a de-centralized approach in response to the governance problems? Journal of the American Medical Informatics Association 26, 76–80.

[67] Simmons, C., Ellis, C., Shiva, S., Dasgupta, D., Wu, Q., 2014. Avoidit: A cyber attack taxonomy, in: 9th Annual Symposium on Information Assurance (ASIA'14), pp. 2–12.

[68] Stephens, Z.D., Lee, S.Y., Faghri, F., Campbell, R.H., Zhai, C., Efron, M.J., Iyer, R., Schatz, M.C., Sinha, S., Robinson, G.E., 2015. Big data: astronomical or genomical? PLoS biology 13, e1002195.

[69] Suter, S.M., 2007. A brave new world of designer babies. Berkeley Tech. LJ 22, 897.

[70] Sweeney, L., 2000. Simple demographics often identify people uniquely. Health (San Francisco) 671, 1–34.

[71] Sweeney, L., Abu, A., Winn, J., 2013. Identifying participants in the personal genome project by name (a re-identification experiment). arXiv preprint arXiv:1304.7605 .

[72] Tao, T., Chen, Y., Liu, B., Jin, X., Yan, M., Ji, S., 2018. Security analysis of bioinformatics web application, in: International Conference on Security with Intelligent Computing and Big-data Services, Springer. pp. 383–397.

[73] Turner, G., 2019. The growing need for cyberbiosecurity, in: InSITE 2019: Informing Science+ IT Education Conferences: Jerusalem, pp. 207–215.

[74] UCSC, (accessed February 28, 2020). Genome browser. https://genome.ucsc.edu/.

[75] Van Aken, J., Hammond, E., 2003. Genetic engineering and biological weapons. EMBO reports 4, S57–S60.

[76] Vengadapurvaja, A., Nisha, G., Aarthy, R., Sasikaladevi, N., 2017. An efficient homomorphic medical image encryption algorithm for cloud storage security. Procedia computer science 115, 643–650.

[77] Vinatzer, B.A., Heath, L.S., Almohri, H.M., Stulberg, M.J., Lowe, C., Li, S., 2019. Cyberbiosecurity challenges of pathogen genome databases. Frontiers in bioengineering and biotechnology 7.

[78] Wirth, A., 2020. Cyberinsights: Talking about the software supply chain. Biomedical Instrumentation & Technology 54, 364–367.

[79] Zonana, K., . Crispr critters and crispr conundrums. https://scopeblog.stanford.edu/2015/12/03/crispr-critters-and-crispr-conundrums/.