A Vision to Mitigate Bioinformatics Software Development Challenges

Akond Rahman
Tennessee Technological University
Cookeville, Tennessee, USA
arahman@tntech.edu

Farzana Ahamed Bhuiyan
Tennessee Technological University
Cookeville, Tennessee, USA
fbhuiyan42@tntech.edu

ABSTRACT

Developers construct bioinformatics software to automate crucial analysis and research related to biological science. However, challenges while developing bioinformatics software can prohibit advancement in biological science research. Through a human-centric systematic analysis, we can identify challenges related to bioinformatics software development and envision future research directions. From our qualitative analysis with 221 Stack Overflow questions, we identify six categories of challenges: file operations, searching genetic entities, defect resolution, configuration management, sequence alignment, and translation of genetic information. To mitigate the identified challenges we envision three research directions that require synergies between bioinformatics and automated software engineering: (i) automated configuration recommendation using optimization algorithms, (ii) automated and comprehensive defect categorization, and (iii) intelligent task assistance with active and reinforcement learning.

CCS CONCEPTS

• **Software and its engineering** → **Application specific development environments**.

KEYWORDS

bioinformatics, challenge, empirical study, stack overflow

ACM Reference Format:
Akond Rahman and Farzana Ahamed Bhuiyan. 2020. A Vision to Mitigate Bioinformatics Software Development Challenges. In *35th IEEE/ACM International Conference on Automated Software Engineering Workshops (ASEW '20), September 21–25, 2020, Virtual Event, Australia.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3417113.3422155

1 INTRODUCTION

According to the U.S. National Institute of Health (NIH), the domain of bioinformatics is related to research and development of computational methodologies and software for expanding the use of biological and medical data [22, 28]. Bioinformatics software is used to archive and analyze biological and medical data [22, 28].

Bioinformatics software plays a pivotal role in disease diagnostics and prevention [8]. An example bioinformatics software is 'Nextstrain', which is used to analyze SARS-CoV-2, the new coronavirus responsible for COVID-19 [40]. Nextstrain is an open source software (OSS) that is used to perform genetic epidemiology [40], the domain where researchers investigate the role of genetic and environmental factors in disease contamination [23]. Using Nextstrain public health experts are making informed decisions related to travel restrictions, school closures, and lock down orders. Nextstrain is an example of how bioinformatics software can directly impact policy making related to public health and potentially save human lives.

However, challenges unique to bioinformatics software usage may hinder progress in biological science research. These challenges can delay the development and deployment of bioinformatics software, which in turn may prevent the analysis of genetic entities, such as pathogens that cause pandemics similar to COVID-19. As software development is integral to modern-day bioinformatics research [8], identification of challenges related to bioinformatics software usage is of paramount importance. From the identified challenges we can lay out bioinformatics-related research areas for the software engineering research community.

One strategy for challenge identification is to investigate questions posted on Stack Overflow (SO), a popular website, where developers seek solutions to programming-related challenges. Let us consider Figure 1, which shows an example SO question [35], where a bioinformatics developer seeks help to find nucleotides, i.e., organic molecules that form nucleic acids, such as DNA and RNA. In prior work, researchers have documented the value of analyzing SO questions to identify challenges related to static analysis [21], deep learning [18], and privacy [38]. Our hypothesis is that by systematically analyzing SO questions similar to Figure 1, we also can identify what challenges bioinformatics developers encounter.

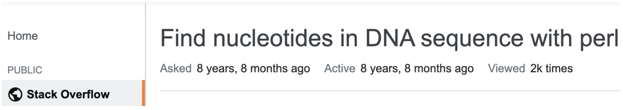
We answer the research question: ***RQ:** What challenges are expressed by bioinformatics software developers on Stack Overflow?*

We conduct an empirical study by applying a qualitative analysis technique called open coding [32] on 221 SO questions related to bioinformatics. Based on our categorization, we propose future research directions related to bioinformatics software development. **Our contribution** is a derived list of challenges related to bioinformatics software development.

2 RELATED WORK

Lawlor and Walsh [25]'s paper is the closest to our paper in spirit. They [25] advocated for training software engineers with biological expertise to deliver high quality bioinformatics software. Our paper

ASEW '20, September 21–25, 2020, Virtual Event, Australia
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-8128-4/20/09…$15.00
https://doi.org/10.1145/3417113.3422155



Find nucleotides in DNA sequence with perl

Asked 8 years, 8 months ago Active 8 years, 8 months ago Viewed 2k times

Figure 1: Example SO question [35]: the SO user is seeking help on finding nucleotides in a DNA sequence.

is also related to prior research that has investigated software development issues for bioinformatics [6, 7, 15, 31, 39]. Russell et al. [31] investigated the nature of contribution in bioinformatics software projects and observed that female contribution decreases as bioinformatics repositories become more and more popular. Chilana et al. [7] studied the differences in programming styles between computer scientists and biologists, and observed that computer scientists tend to focus more on program performance instead of theoretical constructs of biology. Fourment et al. [15] investigated the use of programming languages in bioinformatics software and recommended that developers should choose a programming language by taking performance and library availability into account. Thireou et al. [39] investigated the availability of web resources for developers. Cashman et al. [6] investigated how configuration manipulation can impact behavior of bioinformatics software programs.

The above-mentioned discussion highlights research related to education, empirical studies, human factors, and practices but a lack of research on the challenges unique to bioinformatics software development. Our hypothesis is: unique programming challenges exist for bioinformatics software development, which can be identified by analyzing SO posts.

3 EMPIRICAL STUDY

In this section, we provide the methodology and results for **RQ: What challenges are expressed by bioinformatics software developers on Stack Overflow?**

3.1 Methodology

On SO, users ask questions related to software development [13]. Our hypothesis is that by investigating SO questions we can identify challenges specific to bioinformatics software development. Each question posted on SO has a title that provides a concise summary of what the question is about [13]. The details of the question are presented in the body, where users can describe the problem in detail with additional references [13]. Each question has one or many tags, which are used to identify the applicable language or technology for the question. Scores in SO questions are indicative of quality [3].

Dataset: We use the SOTorrent dataset [4] created on March 15, 2020. First, we identify SO questions with the tag *'bioinformatics'* to extract bioinformatics-related SO questions. According to prior work [14, 29], SO datasets suffer from quality issues. Similar to prior research [14], we apply a filtering criteria to improve the quality of the downloaded data, which is summarized in Table 1. Altogether, we collect 221 SO questions to answer our research question.

Table 1: Selection of Bioinformatics-related SO Questions

Initial question count	41,782,536
Criteria-1 (Questions tagged as 'bioinformatics')	2,712
Criteria-2 (Questions with at least one answer)	2,380
Criteria-3 (Questions with score > 0)	1,377
Criteria-4 (Questions that describe challenges)	221
Final question count	221

Open coding: We apply open coding [32] on the collected 221 SO questions. In open coding, a rater observes and synthesizes patterns within unstructured text [32]. First, we read the question description and title to obtain raw text, which is merged into codes. Next, we merged the codes based on similarities to derive categories.

The first author derived the categories, which are susceptible to bias. We verify the first author's rating by allocating another rater, who is the last author of the paper. The last author applied closed coding [11] on a randomly selected set of 50 SO questions. For each of the 50 SO questions, the last author examined if the question maps to any of the categories identified by the first author. We calculate the agreement rate using Cohen's Kappa [10].

3.2 Findings

We identify six categories of development challenges for bioinformatics. We describe each category with examples that are presented in the ($_{PID}$) format, where *PID* is the ID of the SO question. The count of questions that map to each challenge is enclosed within parenthesis. For example, 91 of the studied 221 questions belong to the category 'File Operations'.

I. *File Operations* (91): This category is related to processing files that store bioinformatics data. Examples of file processing operations include reading and writing bioinformatics storage files, such as BAM, FASTA, and SMILES [2]. For example, we observe a SO user to ask how to read the content from a SMILES file, which is used to store biomolecules ($_{14826373}$). Conversion of one file type to another file type can also be challenging: a developer sought help on how to convert the content of a FASTA file to an ALN file, which is typically used to store sequence alignment ($_{23881801}$). The FASTA file is a text-based format for representing nucleotide or amino acid sequences using single-letter codes and is considered as the defacto standard to store nucleotide sequence data [27]. We observe the question creators to be unaware of existing APIs, such as 'Open Babel' [1] to read SMILES files, and 'Bio.Python.Applications' [9] to convert FASTA files.

II. *Searching Genetic Entities* (73): This category is related to searching for genetic entities, such as k-mers and motifs. For this category, developers search for a certain entity within gene sequence data, which are often represented as strings or sets of strings. One example is searching for k-mers within multiple DNA sequences ($_{31769943}$). k-mers are sub-sequences of length k contained within a DNA/RNA sequence. As another example, we observe a SO user to search for motifs, i.e., sub-sequences of a protein or a DNA sequence that has a specific structure ($_{22809820}$).

III. *Defect Resolution* (20): This category is related to understanding the causes of defects in bioinformatics software and mitigating such defects. While constructing bioinformatics software programs, developers encounter defects related to dependencies ($_{16199037}$), faults ($_{26592680}$) and performance ($_{24378495}$). Developers seek

[1] http://openbabel.org/wiki/Main_Page

guidance on how to troubleshoot and mitigate such defects. In response, the SO community provides clues on why a defect is occurring, and code constructs to mitigate such defects. For example, while writing a computer program to understand gene expression and drug interactions a SO user experienced program crashes ($_{24378495}$). Inefficient allocation of arrays attributed to the crash. As another example, a corrupted payload downloaded from the Internet was responsible for a dependency defect ($_{16199037}$).

IV. *Configuration Management of Bioinformatics Software* (18): The category is related to configuration management of bioinformatics software. When using bioinformatics software developers are not aware of existing configurations that will provide optimal results to accomplish a certain task. For example while using BLAST [1], a software used to search for gene sequences, a developer was unaware of existing BLAST configurations that can help to customize search results ($_{1778193}$). Our observations are supported by Cashman et al. [6] who also observed developers to struggle with configurations of bioinformatics software.

V. *Sequence Alignment* (14): This category is related to sequence alignment operations in bioinformatics. Sequence alignment is the technique of arranging the sequences of a DNA, RNA, or a protein to identify regions that are similar with respect to structure or functionality [26]. Sequence alignment is used to identify homologous genes i.e., genes that share common ancestors. From our analysis, we notice developers to seek help when implementing an algorithm to perform sequence alignment in Python ($_{23400317}$) and R ($_{4497747}$). We also notice developers to ask about appropriate data structures while implementing sequence alignment algorithms ($_{21199263}$).

VI. *Translation of Genetic Information* (5): This category is related to transforming one category of genetic information to another category of genetic information. We notice developers to miss existing libraries, such as BioStrings [20] when attempting to convert one category of genetic information to another ($_{42986106}$). For example, a developer was not aware about BioStrings, and asked how to create a mapping between codons and amino acids ($_{42986106}$). A codon is a 3-letter code, which corresponds to a specific amino acid [16]. Each letter in a codon corresponds to a chemical compound [16]. For example, for the nucleotide sequence *ACUACGGAG* the codons are *ACU*, *ACG*, *GAG*. The letters *A*, *C*, *G*, and *U* respectively, correspond to four chemical compounds, Adenine, Cytosine, Guanine, and Uracil.

Rater Verification: The Cohen's Kappa is 0.72 indicating 'substantial' agreement [24].

Limitations: We discuss the limitations of our paper below:

- Our empirical study is based on the content of 221 SO questions, which is susceptible to external validity. Developers also ask questions on other forums, such as BioStars [2].
- The six categories of questions are derived by the first author, and therefore the derivation process is susceptible to rater bias.

4 VISION

We identify three future research directions:

I. *Configuration Research*: We have documented evidence in Section 3.2 on challenges related to configuration management. We advocate for research that can investigate how developers perceive configuration management of bioinformatics software and the reasons that attribute to such perception. Identified reasons can be leveraged in constructing developer-friendly configuration tools and documentation. Furthermore, we advocate for novel research that will automatically tune and recommend configuration settings well-suited for the development task of interest. For example, if a developer wants optimal program performance, then the objective will program execution time, for which the algorithm will find the appropriate combinations of configuration values. Optimization algorithms, such as differential evolution and sequential model-based optimization [12] can be applied.

II. *Defects Research*: As shown in Section 3.2, defect resolution is one challenge for which developers seek help on. We observe anecdotal evidence related to dependency and performance defects, which must be substantiated through systematic empirical investigation. We also hypothesize other defect categories to exist. To investigate other defect categories researchers can investigate other data sources, such as OSS repositories hosted on GitHub, along with questions posted on BioStars and SO. In prior work [30, 41], researchers have documented the importance of defect categorization. Defect categorization for bioinformatics software can help in (i) understanding the nature of defects, (ii) constructing automated defect categorization, detection, and repair tools, and (iii) measuring bioinformatics software quality.

III. *Intelligent Learning to Assist Developers*: From our discussion in Section 3.2 we observe developers to be unaware of existing APIs to accomplish certain tasks, such as file operations, searching genetic entities, sequence alignment, and translation of genetic information. We hypothesize that developers are not always aware of available APIs and code snippets that can help developers to accomplish necessary tasks. We envision research that will substantiate our hypothesis related to developer awareness on existing APIs. Future research should also rigorously evaluate how existing automated API recommendation techniques, such as Biker [19] and DeepAPI [17], as well as code recommendation techniques, such as NLP2Code [5] can be adapted for bioinformatics software. Existing research can complement novel research pursuits that will investigate how relevant APIs and code snippets can be recommended automatically. Our vision is to create a 'just-in-time' technique to recommend necessary libraries and code snippets, which will predict the task the developer is trying to accomplish, and generate recommendations accordingly. Learning techniques, such as reinforcement learning [36] and active learning [33] can be leveraged to continuously learn from developer-written source code. In active learning, the algorithm queries an information source [33]. The information source could be a developer accomplishing a certain task. Reinforcement learning uses a reward-based approach [36]. Developers' acceptance or rejection of a code snippet or API recommendation while accomplishing a task can be used to formulate rewarding in reinforcement learning. In other domains, such as for web recommendation systems [37] and intelligent tutoring systems [34], researchers have reported the usefulness of reinforcement learning and active learning techniques.

[2] https://www.biostars.org/

Conclusion: Despite advancements in bioinformatics software, developers face challenges that can hinder research in biological science. We have derived a list of six challenges that developers have reported on SO related to bioinformatics software development. Based on our identified challenges, we envision three research directions that require strong synergies between automated software engineering and bioinformatics.

ACKNOWLEDGMENTS

We thank the PASER group at Tennessee Tech. University for their valuable feedback. The research was partially funded by the Cybersecurity Education, Research and Outreach Center (CEROC) at Tennessee Tech. University.

REFERENCES

[1] Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. 1990. Basic local alignment search tool. *Journal of molecular biology* 215, 3 (1990), 403–410.

[2] Evan Anderson, G. Veith, and David Weininger. 1987. SMILES: a line notation and computerized interpreter for chemical structures.

[3] P. Arora, D. Ganguly, and G. J. F. Jones. 2015. The good, the bad and their kins: Identifying questions with negative scores in StackOverflow. In *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. 1232–1239.

[4] Sebastian Baltes, Lorik Dumani, Christoph Treude, and Stephan Diehl. 2018. SOTorrent: Reconstructing and Analyzing the Evolution of Stack Overflow Posts. In *Proceedings of the 15th International Conference on Mining Software Repositories* (Gothenburg, Sweden) *(MSR '18)*. ACM, New York, NY, USA, 319–330. https://doi.org/10.1145/3196398.3196430

[5] Brock Angus Campbell and Christoph Treude. 2017. NLP2Code: Code snippet content assist via natural language tasks. In *2017 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 628–632.

[6] Mikaela Cashman, Myra B. Cohen, Priya Ranjan, and Robert W. Cottingham. 2018. Navigating the Maze: The Impact of Configurability in Bioinformatics Software. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering* (Montpellier, France) *(ASE 2018)*. Association for Computing Machinery, New York, NY, USA, 757–767. https://doi.org/10.1145/3238147.3240466

[7] P. K. Chilana, C. L. Palmer, and A. J. Ko. 2009. Comparing bioinformatics software development by computer scientists and biologists: An exploratory study. In *2009 ICSE Workshop on Software Engineering for Computational Science and Engineering*. 72–79.

[8] Levin Clement, Dynomant Emeric, Mouchard Laurent, Landsman David, Hovig Eivind, Vlahovicek Kristian, et al. 2018. A data-supported history of bioinformatics tools. *arXiv preprint arXiv:1807.06808* (2018).

[9] Peter J. A. Cock, Tiago Antao, Jeffrey T. Chang, Brad A. Chapman, Cymon J. Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, and Michiel J. L. de Hoon. 2009. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* 25, 11 (03 2009), 1422–1423. https://doi.org/10.1093/bioinformatics/btp163 arXiv:https://academic.oup.com/bioinformatics/article-pdf/25/11/1422/944180/btp163.pdf

[10] Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement* 20, 1 (1960), 37–46. https://doi.org/10.1177/001316446002000104

[11] Benjamin F Crabtree and William L Miller. 1999. *Doing qualitative research.* sage publications.

[12] Kalyanmoy Deb. 2001. *Multi-objective optimization using evolutionary algorithms*. Vol. 16. John Wiley & Sons.

[13] Stack Exchange. 2019. Stack Exchange. https://data.stackexchange.com/. [Online; accessed 08-06-2020].

[14] E. Farhana, N. Imtiaz, and A. Rahman. 2019. Synthesizing Program Execution Time Discrepancies in Julia Used for Scientific Software. In *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. 496–500.

[15] Mathieu Fourment and Michael R. Gillings. 2007. A comparison of common programming languages used in bioinformatics. *BMC Bioinformatics* 9 (2007), 82 – 82.

[16] Anthony JF Griffiths, Susan R Wessler, Richard C Lewontin, William M Gelbart, David T Suzuki, Jeffrey H Miller, et al. 2005. *An introduction to genetic analysis*. Macmillan.

[17] Xiaodong Gu, Hongyu Zhang, Dongmei Zhang, and Sunghun Kim. 2016. Deep API Learning. In *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering* (Seattle, WA, USA) *(FSE 2016)*. Association for Computing Machinery, New York, NY, USA, 631–642. https://doi.org/10.1145/2950290.2950334

[18] Junxiao Han, Emad Shihab, Zhiyuan Wan, Shuiguang Deng, and Xin Xia. 2020. What do Programmers Discuss about Deep Learning Frameworks. *EMPIRICAL SOFTWARE ENGINEERING* (2020).

[19] Qiao Huang, Xin Xia, Zhenchang Xing, David Lo, and Xinyu Wang. 2018. API Method Recommendation without Worrying about the Task-API Knowledge Gap. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering* (Montpellier, France) *(ASE 2018)*. Association for Computing Machinery, New York, NY, USA, 293–304. https://doi.org/10.1145/3238147.3238191

[20] Wolfgang Huber, Vincent J Carey, Robert Gentleman, Simon Anders, Marc Carlson, Benilton S Carvalho, Hector Corrada Bravo, Sean Davis, Laurent Gatto, Thomas Girke, et al. 2015. Orchestrating high-throughput genomic analysis with Bioconductor. *Nature methods* 12, 2 (2015), 115.

[21] Nasif Imtiaz, Akond Rahman, Effat Farhana, and Laurie Williams. 2019. Challenges with Responding to Static Analysis Tool Alerts. In *Proceedings of the 16th International Conference on Mining Software Repositories* (Montreal, Canada) *(MSR '19)*.

[22] Someswa Kesh and Wullianallur Raghupathi. 2004. Critical issues in bioinformatics and computing. *Perspectives in health information management/AHIMA, American Health Information Management Association* 1 (2004).

[23] Muin J Khoury, Terri H Beaty, Terri H Beaty, Bernice H Cohen, et al. 1993. *Fundamentals of genetic epidemiology*. Vol. 22. Monographs in Epidemiology and.

[24] J. Richard Landis and Gary G. Koch. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics* 33, 1 (1977), 159–174. http://www.jstor.org/stable/2529310

[25] Brendan Lawlor and Paul Walsh. 2015. Engineering bioinformatics: building reliability, performance and productivity into bioinformatics software. *Bioengineered* 6, 4 (2015), 193–203. https://doi.org/10.1080/21655979.2015.1050162 arXiv:https://doi.org/10.1080/21655979.2015.1050162 PMID: 25996054.

[26] David W Mount. 2001. *Bioinformatics: sequence and genome analysis*. Vol. 1. Cold spring harbor laboratory press New York.

[27] NCBI. 2020. BLAST Topics. https://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Web&PAGE_TYPE=BlastDocs&DOC_TYPE=BlastHelp [Online; accessed 09-06-2020].

[28] NCBI. 2020. National Center for Biotechnology Information. https://www.ncbi.nlm.nih.gov/. [Online; accessed 07-06-2020].

[29] Akond Rahman, Effat Farhana, and Nasif Imtiaz. 2019. Snakes in Paradise?: Insecure Python-related Coding Practices in Stack Overflow. In *Proceedings of the 16th International Conference on Mining Software Repositories* (Montreal, Canada) *(MSR '19)*.

[30] Akond Rahman, Effat Farhana, Chris Parnin, and Laurie Williams. 2020. Gang of Eight: A Defect Taxonomy for Infrastructure As Code Scripts. In *Proceedings of the 42nd International Conference on Software Engineering* (Seoul, South Korea) *(ICSE '20)*. to appear. pre-print: https://akondrahman.github.io/papers/icse20_acid.pdf.

[31] Pamela H. Russell, Rachel L. Johnson, Shreyas Ananthan, Benjamin Harnke, and Nichole E. Carlson. 2018. A large-scale analysis of bioinformatics code on GitHub. *PLOS ONE* 13, 10 (10 2018), 1–19. https://doi.org/10.1371/journal.pone.0205898

[32] Johnny Saldana. 2015. *The coding manual for qualitative researchers*. Sage.

[33] Burr Settles. 2009. *Active learning literature survey.* Technical Report. University of Wisconsin-Madison Department of Computer Sciences.

[34] Yi Shang, Hongchi Shi, and Su-Shing Chen. 2001. An Intelligent Distributed Environment for Active Learning. *J. Educ. Resour. Comput.* 1, 2es (Aug. 2001), 4–es. https://doi.org/10.1145/384055.384059

[35] Stack Overflow. 2011. bioinformatics - Find nucleotides in DNA sequence with perl. https://stackoverflow.com/questions/7090371/. [Online; accessed 06-06-2020].

[36] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction.* MIT press.

[37] Nima Taghipour, Ahmad Kardan, and Saeed Shiry Ghidary. 2007. Usage-Based Web Recommendations: A Reinforcement Learning Approach. In *Proceedings of the 2007 ACM Conference on Recommender Systems* (Minneapolis, MN, USA) *(RecSys '07)*. Association for Computing Machinery, New York, NY, USA, 113–120. https://doi.org/10.1145/1297231.1297250

[38] Mohammad Tahaei, Kami Vaniea, and Naomi Saphra. 2020. Understanding Privacy-Related Questions on Stack Overflow. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376768

[39] Trias Thireou, George Spyrou, and Vassilis Atlamazoglou. 2007. A Survey of the Availability of Primary Bioinformatics Web Resources. *Genomics, Proteomics Bioinformatics* 5, 1 (2007), 70 – 76. https://doi.org/10.1016/S1672-0229(07)60017-5

[40] Emily Waltz. 2020. Software and Genetic Sequencing Track the Coronavirus's Path. https://spectrum.ieee.org/the-human-os/biomedical/devices/genetic-sequencing-and-online-software-tools-track-caronaviruss-path. [Online; accessed 07-05-2020].

[41] Zhiyuan Wan, David Lo, Xin Xia, and Liang Cai. 2017. Bug Characteristics in Blockchain Systems: A Large-Scale Empirical Study. In *2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)*. 413–424.