Interpreting and integrating big data in the life sciences

Serghei Mangul[12]

[1]Department of Computer Science, University of California Los Angeles, Los Angeles, CA, USA

[2]Institute for Quantitative and Computational Biosciences, University of California Los Angeles, Los Angeles, CA, USA

Email: smangul@ucla.edu

Abstract (200 words)

Recent advances in omics technologies have led to the broad applicability of computational techniques across various domains of life science and medical research. These technologies provide an unprecedented opportunity to collect omics data from hundreds of thousands of individuals and to study gene-disease association without the aid of prior assumptions about the trait biology. Despite the many advantages of modern omics technologies, interpretations of big data produced by such technologies require advanced computational algorithms. Below I outline key challenges that biomedical researches are facing when interpreting and integrating big omics data. I discuss the reproducibility aspect of big data analysis in the life sciences and review current practices in reproducible research. Finally, I explain the skills which biomedical researchers need to acquire in order to independently analyze big omics data.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27603v2 | CC BY 4.0 Open Access | rec: 7 Jun 2019, publ: 7 Jun 2019

Introduction

Recent advances in omics technologies have led to the broad applicability of modern high-throughput technologies across various domains of life science and medical research. These technologies are capable of generating big-data sets across large-scale clinical cohorts allows connecting complex diseases to relevant genomic features. However, analysis of big data requires the use of sophisticated bioinformatics algorithms capable of differentiating technical noise from the biological signal in the data. The analysis of big data in life sciences typically starts with the analysis of raw data and concludes with data visualization and interpretation of patterned data produced by analyses (**Figure 1**). Big data represented by massive datasets represent a substantial challenge for the analysis due to an increased computational footprint associated with handling, processing, and moving information[1,2]. There are various definitions of big data across various domains of science, ranging from a simple definition that a big data is a data which is too large and complex to be processed using traditional non-computational approaches [3]. More complex definitions of big data require several important features to be present in the data before it can be classified as big data. For example, a popular 3V definition requires volume, variety, and velocity of the data[3,4]. Applying computational methods to the big data will provide the power to make novel biological, translational and clinical discoveries and push the boundaries of current knowledge of the biology of disease, as well as phenotypic and clinical dynamics of the disease. The key foundational aspects are bioinformatics methods, which allow to extract relevant biological

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27603v2 | CC BY 4.0 Open Access | rec: 7 Jun 2019, publ: 7 Jun 2019

signal from noisy datasets and eventually enable discovery, translation and actionable applications. Modern biomedical data sets contain tens of thousands of samples and petabytes of raw data (e.g TCGA[5], GTEx[6]) and typically satisfy the definition of big data and are considered big biomedical data[6–8]. In contrast with the raw biomedical and sequencing data, summary statistics extracted from such datasets (e.g recount2[9]) are significantly smaller and require less computational resources to be processed and analyzed. In this review, I will discuss the raw biomedical data and challenges and opportunities associated with processing and analyzing such datasets. I will also discuss the computational skills required for analysis and interpretation of big omics data. These skills include the ability to operate command line and run bioinformatics directly on high clusters.

Acquiring such skills can be challenging to many life science and medical researchers. At present, the bioinformatics community lack mechanisms to researchers in effectively learning to analyze big omics data.

Ability to leverage various types of omics datasets from large-scale clinical cohorts is essential to studying the functional mechanisms underlying the connections between genetics, immune system, and disease etiology. For example, the availability of rich omics data generated by the TCGA consortium[5] provides an exciting opportunity for the discovery of how genetic variation affects the development, progression, and drug response of cancerous tumors[10]. Despite many studies using TCGA data for the basic research[11], the translational value of such datasets yet to be fully unlocked [12].

Additionally, bioinformatics methods for large-scale clinical cohorts promise to identify novel markers prognostic of disease risk across a variety of diseases, including cancer.

Despite the increasing size and complexity of datasets in the biological and medical sciences, many biomedical researchers today lack sufficient computational skills to analyze the large-scale data they generate. At present, the digital gap in contemporary biology limits the potential of these biomedical researchers to creatively explore their data. Further, the digital gap limits the collaborative potential of these biomedical researchers and computational scientists[13]. Training of life sciences in computational techniques can potentially narrow this digital gap, expand the skills of biomedical researchers, and improve the ability of biomedical researchers to leverage the consultation services they seek with computational scientists. Such training should include both hands-on sessions as well as session covering the principle of computational methods. Especially the assumption and heuristic of bioinformatics methods. Conceptual understanding of bioinformatics algorithms will allow biomedical researchers to better understand and interpret the bioinformatics results. Such skills as automating tasks with the Unix command line are necessary to process big data on high-performance data. Knowledge of Python or R is required to analyze the data on the laptop and visualize the obtained results.

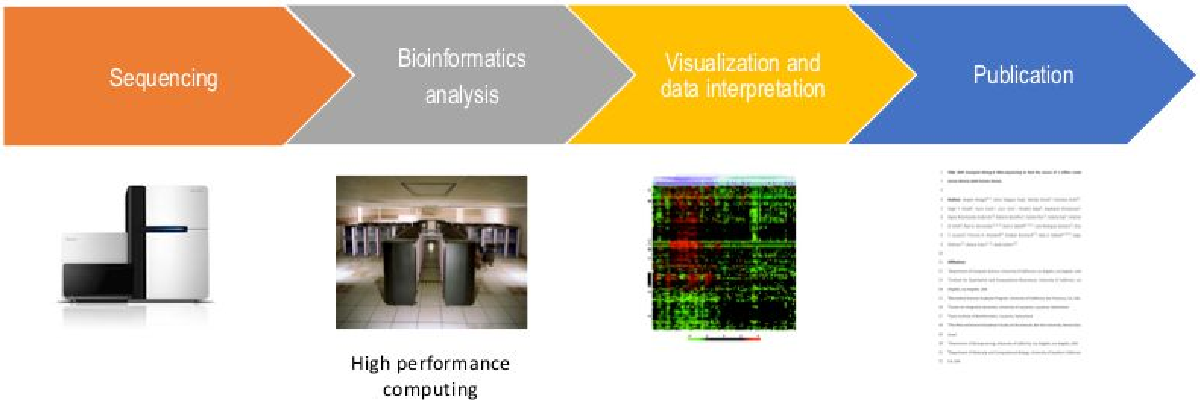
Sequencing
Bioinformatics analysis
Visualization and data interpretation
Publication
High performance computing

Figure 1. Workflow of big data analysis and interpretation in the life sciences.

Bioinformatics methods in life science research

During the past decade, the rapid advancement of omics technologies has led to the development of an enormous amount and diversity of bioinformatics algorithms across various fields of modern biology[14]. hen applied to high-dimensional clinical datasets, bioinformatics algorithms identification of novel disease subtypes and discovery of novel markers which may be prognostic of disease risk. Such bioinformatics algorithms are usually encapsulated as computational software tools[15]. The majority of bioinformatics tools are designed for UNIX operating system, which requires a user to operate the tools using command line--without the benefit of a graphical user interface (GUI). Additionally, UNIX framework makes it possible to connect different bioinformatics tools to communicate without having been designed explicitly to work together using pipes. This also allows avoiding the creation of unnecessary temporary files for each bioinformatics tool of the pipeline

Barriers in interpreting and integrating big data in the life sciences

A major barrier in interdisciplinary studies is a lack of a common communication style between researchers trained and working in increasingly specialized academic disciplines[16]. Today's big-data projects require that life science researchers either learn how to use command-line tools or outsource their data analysis to computational experts. Active engagement in analyzing data generated by life science researchers is essential to advancing interdisciplinary research in the biological sciences. However, biologists and medical researchers often lack formal training in the use of computational techniques. Scholars have developed teaching models aimed to support biomedical researchers transition from using a graphical interface (e.g., Microsoft Excel) to UNIX command line[13,17,18]. Biomedical researchers often possess various levels of computational skills; workshops require adjusting the teaching pace for trainees who have different levels of computational background[19]. Additionally, the ability of trainees to switch from a graphical interface with the assistance of a mouse to a command line interface with no mouse support varies across the life science researchers.

Training a life science research to use computational techniques poses unique challenges and requires a special approach. For example, such training does not require cultivating a deep understanding of fundamental computer science principles. Instead, such training is limited to applied learning, requiring quick assimilation of introduced techniques and acquired skills within the context of the research project of interest. In general, flexible pedagogy is preferred. For example, instead of introducing the fundamental computational concepts formally, such concepts are introduced on an as-needed basis, are combined with numerous hands-on examples, and rely on the instructor's guidance to consolidate the learner's newly-acquired knowledge.

Biomedical researchers who have completed the training in computational skills often continue to engage with the command line and with training instructors long after the training session has formally ended. Researchers who have completed the workshops are also able to better engage with computer scientists when seeking consultation services for their projects. The training model I developed [13] has helped life science researchers to learn the "language of computing," a skill that allows them to better understand what analyses can be accomplished with UNIX, and how to ask for specific types of help from computational scientists.

Alternatively, biomedical researchers can delegate large-scale data analyses to bioinformatics cores. However, outsourcing analyses present several challenges. Many complex issues arise during the analysis of big omics data which are difficult to predict in advance. In such cases, life science and biomedical research can optimize the analysis if they are adequately trained and remain involved in the analytical workflow. In addition, research groups utilizing the core often want to move the project in different directions from what was originally proposed. Another approach is to develop a GUI that allows researchers with a limited computational background to easily create, run, and troubleshoot analytical pipelines. While useful to researchers with a limited computational background, these interfaces may have limited computational capacity compared to high-performance clusters and might not suitable for analysis of big omics data generated from some large-scale clinical cohorts.

Computational resources required to analyze the big omics data

Computational resources required to analyze the big omics data differs significantly depending on the step of the analysis. Analyses dealing with raw data typically require a significant amount of computational resources to perform essential tasks and space to store the output data. Analyzing big omics data can typically be performed only on high-performance clusters. For example, the analysis of differentially expressed genes based on RNA-Seq data starts with raw reads produced by the sequencing machines and concludes with a list of differentially expressed genes with corresponding gene expression fold changes. The computational resources and amount of storage needed to analyze and store the results of analysis significantly various across the various step of the analysis. For example, the total space required to store raw sequencing data of one sample is approximately 15G (**Figure 2)**. It is possible to store such data on a regular workstation with no need for a large HPC environment. However, the increasing size of the cohorts composed of thousands of individuals makes storing data on a regular workstation impractical. Additionally, the regular workstation lacks the computational power to process thousands of samples. As a result, the steps of the analysis requiring significant computational resources and space to store the data can be only performed on high-performance clusters or using cloud computing. Decreasing cost of cloud computing makes it an attractive alternative to well established high-performance clusters. The exact cost of cloud computing is constantly decreasing and is compared to high-performance cluster elsewhere[20].

Other steps require a smaller amount of resources and thus can perform locally on the personal computer or laptop. Typically, the analysis performed on the local machine does not require the knowledge of command line skills. Such analyses typically involve various statistical analyses and visualization steps, and these tasks can be performed using the widely popular statistical language *R*[21]. For example, once the gene expression levels were obtained from RNA-Seq data on the high-performance cluster, one can transfer them to a personal computer and locally perform differential expression analysis using available *R* packages[22].

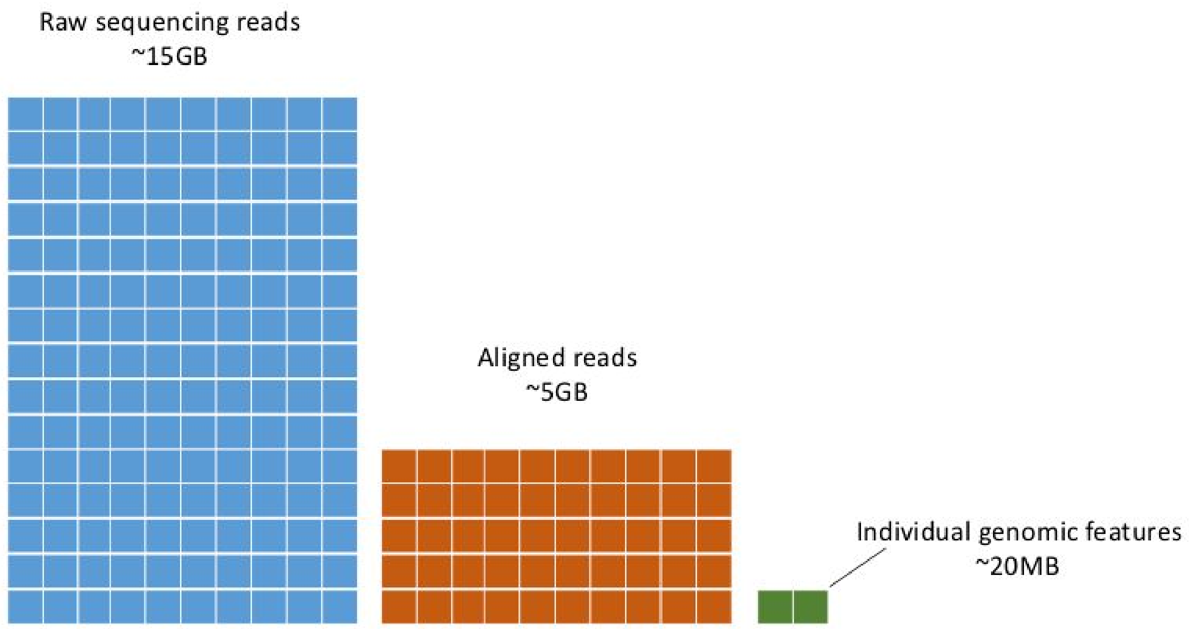

Figure 2. The amount of space needed to store the results of the sequencing analysis. The size of each step is shown as a grid of boxes, each box is equivalent to storage of 10 Mb.

Computational reproducibility in life science research

An astonishing number of bioinformatics software tools are designed to accommodate increasingly bigger, more complex, and more specialized bio-datasets are developed each year[14]. With the increasing importance and popularity of computational and data-enabled approaches among biomedical researchers, it becomes ever more critical to ensure that the developed software is usable[23] and the Uniform Resource Locator (URL), through which the software tool is accessible, is archivally stable. Consistently usable and accessible software provides a foundation for the reproducibility of published biomedical research, defined as the ability to replicate published findings by running the same computational tool on data generated by the study[23,24].

Open data, open software, and reproducible research are important aspects of big data analysis in the life sciences[25]. Reproducing previously published results can be made possible, by releasing all research objects, such as raw data, and publically available, archivally stable, and installable computer code. However, a lack of strict implementation or enforcement of journal policies for resource sharing harms rigor and reproducibility as some authors refuse to share the data[26] or source code. Even when the code and data are shared, it still can be challenging to computationally reproduce the results of the published paper[27]. One technique to enable computational reproducibility is literate programming, allowing the reader to understand how the research results were obtained by generating the documents that include the code, narratives, and the outputs including figures and tables. One such platform able to mix code with accompanying documentation and text notes is Jupyter[28]. This popular platform allows the reader to follow the documentation, run the code, visualize results in a single notebook usually opened in the browser[29]. Additionally, containers and virtual machines allow to avoid installability issues and run instantly the code across various operating systems and environments. Example of containers includes Docker[30], Vagrant[30,31], and Singularity[32]. A recent case study proposed an example of documentation and tutorials allowing to easily reproduce results using Jupyter/IPython notebook or a Docker container [29]. Other technique and methodologies allowing computational reproducibility are discussed elsewhere[33].

Despite these challenges, consistently usable and accessible software provides a necessary foundation for rigorous and reproducible data-intensive biomedical research[26,34]. In addition, the usability — or, 'user-friendliness' — of software tools is important, and it can affect its scientific utility. Currently, an estimated 74% of computational software resources are accessible through URLs published in the original paper[35]. Many developed tools are difficult to install and some are impossible to install[35]. Kumar and Dudley warn that poorly maintained or implemented tools will hinder progress in "big data"-driven fields[36].

Any journals now require that omics data generated by the published study should be shared when the paper is released, an important step forward toward improving computational reproducibility in our field. However, the bioinformatics community still lacks the comprehensive policies on precisely how openly shared code used to perform the analysis and generate the figures should be. In a promising effort from *eLife* journal, editors suggest that *R* code used to generate figures should be shared together with the figure[37].

Discussion

High-throughput technologies have changed the landscape of training, research, and education in biomedical fields[38]. Big data generated by those technologies across large-scale clinical cohorts can potentially enable a researcher to connect complex diseases to relevant omics features. As our knowledge of scientifically-validated disease-trait matches increase, new opportunities emerge for development of novels diagnostic and therapeutic tools. However, analysis of big data requires the use of sophisticated bioinformatics algorithms which are often packaged as command-line-driven software tools. A researcher who wishes to use such tools must acquire specific computational skillsets, which are not included in the traditional life science curriculum at major Universities. With the increasing size and complexity of big omics datasets in the biological and medical sciences, researchers are facing a growing dilemma of devoting the time to acquire key computational skills or outsource the analyses to computational researchers.

At present, biomedical researchers are not involved in computational training on a large-scale worldwide. In this review, I provide evidence that the computational training model - that is, when life science research groups receive training and resources to analyze the data that they generate, is a more sustainable approach. The computational training model of life science researchers, when successfully applied across many research institutions worldwide, has the potential to change the landscape of contemporary biomedical research, training, and education. If a critical mass of biomedical researchers obtains computational skills sufficient for analyzing big data, computational training will more likely become an integral part of analysis curricula at these institutions.

The computational training model offers benefits for both individual researchers and the scientific community. Life science and biomedical researchers gain a competitive skill when learning to conduct analysis in a command-line setting. Today’s omics data generates file sizes too large to be opened on a personal computer. These novice computational researchers often must perform their analyses on a high-performance cluster with command line tools and, in the process, the researchers become familiar with programming and basic system administration tasks. Such valuable skills could be leveraged to further the researcher's projects and career.

One important outcome of a comprehensively implemented computational training model is to improve reproducibility in the big data-driven fields of life science and medical research. The standard for rigorous and reproducible analysis is an emerging topic with multiple initiatives across research groups. The scientific community has identified current challenges to ensuring reproducibility of interpreting and integrating big data analysis in the life sciences[26]. For example, *eLife* journal raised the bar of reproducibility, challenging the traditional static representation of data and results of the analysis (usually in the form of PDF or HTML formats). Instead, eLife now suggests a code-based publication, which enables data and analysis to be fully reproducible by the reader[39,40].

Summary

- Recent advances in omics technologies have led to the broad applicability of computational techniques across various domains of life science and medical research. These technologies provide an unprecedented opportunity to collect omics data from hundreds of thousands of individuals and to study gene-disease association without the aid of prior assumptions about the trait biology.

- Interpreting and integrating big data produced by omics technologies require advanced computational algorithms. Despite the increasing size and complexity of datasets in the biological and medical sciences, many biomedical researchers today lack sufficient computational skills to analyze the large-scale data they generate.
- The computational training model of life science researchers, when successfully applied across many research institutions worldwide, has the potential to change the landscape of contemporary biomedical research, training, and education. If a critical mass of life science researchers obtains computational skills sufficient for analyzing big data, computational training will more likely become an integral part of biomedical education.
- One important outcome of a comprehensively implemented computational training model is to improve reproducibility in the big data-driven fields of life science and medical research.

References

1. Mattmann, C. A. Computing: A vision for data science. *Nature* **493**, 473–475 (2013).
2. Marx, V. The big challenges of big data. *Nature* **498**, 255–260 (2013).
3. Mehta, N. & Pandit, A. Concurrence of big data analytics and healthcare: A systematic review. *Int. J. Med. Inform.* **114**, 57–65 (2018).
4. Mauro, A. D., De Mauro, A., Greco, M. & Grimaldi, M. A formal definition of Big Data based on its essential features. *Library Review* **65**, 122–135 (2016).
5. Cancer Genome Atlas Research Network *et al.* The Cancer Genome Atlas Pan-Cancer analysis project. *Nat. Genet.* **45**, 1113–1120 (2013).

6. Consortium, G. & GTEx Consortium. Genetic effects on gene expression across human tissues. *Nature* **550**, 204–213 (2017).

7. Vivian, J. *et al.* Toil enables reproducible, open source, big biomedical data analyses. *Nat. Biotechnol.* **35**, 314–316 (2017).

8. Siva, N. UK gears up to decode 100,000 genomes from NHS patients. *Lancet* **385**, 103–104 (2015).

9. Collado-Torres, L. *et al.* Reproducible RNA-seq analysis using recount2. *Nat. Biotechnol.* **35**, 319–321 (2017).

10. Thorsson, V. *et al.* The Immune Landscape of Cancer. *Immunity* **48**, 812–830.e14 (2018).

11. Park, Y. & Greene, C. S. A parasite's perspective on data sharing. *Gigascience* **7**, (2018).

12. Perera-Bel, J., Leha, A. & Beißbarth, T. Bioinformatic Methods and Resources for Biomarker Discovery, Validation, Development, and Integration. *Predictive Biomarkers in Oncology* 149–164 (2019). doi:10.1007/978-3-319-95228-4_11

13. Mangul, S., Martin, L. S., Hoffmann, A., Pellegrini, M. & Eskin, E. Addressing the Digital Divide in Contemporary Biology: Lessons from Teaching UNIX. *Trends Biotechnol.* **35**, 901–903 (2017).

14. Wren, J. D. Bioinformatics programs are 31-fold over-represented among the highest impact scientific papers of the past two decades. *Bioinformatics* **32**, 2686–2691 (2016).

15. Altschul, S. *et al.* The anatomy of successful computational biology software. *Nat. Biotechnol.* **31**, 894–897 (2013).

16. Via, A. *et al.* Best practices in bioinformatics training for life scientists. *Brief. Bioinform.* **14**, 528–537 (2013).

17. Schneider, M. V. & Jimenez, R. C. Bioinformatics: scalability, capabilities and training in the data-driven era. *Brief. Bioinform.* (2019). doi:10.1093/bib/bbz053

18. Mariano, D., Martins, P., Helene Santos, L. & de Melo-Minardi, R. C. Introducing Programming Skills for Life Science Students. *Biochem. Mol. Biol. Educ.* **47**, 288–295 (2019).

19. Guerfali, F. Z., Laouini, D., Boudabous, A. & Tekaia, F. Designing and running an advanced Bioinformatics and genome analyses course in Tunisia. *PLoS Comput. Biol.* **15**, e1006373 (2019).

20. Dudley, J. T., Pouliot, Y., Chen, R., Morgan, A. A. & Butte, A. J. Translational bioinformatics in the cloud: an affordable alternative. *Genome Medicine* **2**, 51 (2010).

21. Cornillon, P.-A. *et al. R for Statistics*. (CRC Press, 2012).

22. Love, M. I., Huber, W. & Anders, S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.* **15**, 550 (2014).

23. List, M., Ebert, P. & Albrecht, F. Ten Simple Rules for Developing Usable Software in Computational Biology. *PLoS Comput. Biol.* **13**, e1005265 (2017).

24. Beaulieu-Jones, B. K. & Greene, C. S. Reproducibility of computational workflows is automated using continuous analysis. *Nature Biotechnology* **35**, 342–346 (2017).

25. Murphy, F. Open access, open data, FAIR Data and their implications for life sciences researchers. *Emerging Topics in Life Sciences* **2**, 759–762 (2018).

26. Stodden, V., Seiler, J. & Ma, Z. An empirical analysis of journal policy effectiveness for computational reproducibility. *Proc. Natl. Acad. Sci. U. S. A.* **115**, 2584–2589 (2018).

27. Kenall, A. *et al.* Better reporting for better research: a checklist for reproducibility. *Gigascience* **4**, 32 (2015).

28. Project Jupyter. Available at: https://www.jupyter.org. (Accessed: 27th May 2019)

29. Kim, Y.-M., Poline, J.-B. & Dumas, G. Experimenting with reproducibility: a case study of robustness in bioinformatics. *Gigascience* **7**, (2018).

30. Enterprise Application Container Platform | Docker. *Docker* Available at: https://www.docker.com/. (Accessed: 27th May 2019)

31. Introduction - Vagrant by HashiCorp. *Vagrant by HashiCorp* Available at: https://www.vagrantup.com/intro/index.html. (Accessed: 27th May 2019)

32. Singularity | Singularity. Available at: https://singularity.lbl.gov/. (Accessed: 27th May 2019)

33. Piccolo, S. R. & Frampton, M. B. Tools and techniques for computational reproducibility. *GigaScience* **5**, (2016).

34. Beaulieu-Jones, B. K. & Greene, C. S. Reproducibility of computational workflows is automated using continuous analysis. *Nat. Biotechnol.* **35**, 342–346 (2017).

35. Mangul, S. *et al.* A comprehensive analysis of the usability and archival stability of omics computational tools and resources. (2018). doi:10.1101/452532

36. Kumar, S. & Dudley, J. Bioinformatics software for biologists in the genomics era. *Bioinformatics* **23**, 1713–1717 (2007).

37. Bauer, P. C. Writing a Reproducible Paper in R Markdown. *SSRN Electronic Journal* doi:10.2139/ssrn.3175518

38. Hayden, E. C. Genome researchers raise alarm over big data. *Nature* (2015). doi:10.1038/nature.2015.17912

39. Perkel, J. M. Pioneering 'live-code' article allows scientists to play with each other's results.

Nature (2019). doi:10.1038/d41586-019-00724-7

40. Burgess, S. Reproducible research article collection. (2018). doi:10.14293/s2199-1006.1.sor-uncat.clsuuhc.v1