CellPress
REVIEWS

Scientific Life

Addressing the Digital Divide in Contemporary Biology: Lessons from Teaching UNIX

Serghei Mangul,[1,2,*,†] Lana S. Martin,[1,†] Alexander Hoffmann,[3] Matteo Pellegrini,[4] and Eleazar Eskin[1,5]

Life and medical science researchers increasingly rely on applications that lack a graphical interface. Scientists who are not trained in computer science face an enormous challenge analyzing high-throughput data. We present a training model for use of command-line tools when the learner has little to no prior knowledge of UNIX.

The increasing amount of data generated by high-throughput genomics is reshaping the landscape of contemporary life science and medical research into a data science [1]. More life-science and medical research is performed *in silico* [2], and researchers increasingly rely on applications that lack a graphical interface (GUI) and require typing commands through a terminal, generally referred to as the command-line interface (Box 1). The command-line interface has origins in early remote communication devices, such as teleprinter machines, and remains the most efficient computational system. Today's high-throughput data requires applying computational tasks with the command line and represents an enormous challenge to scientists not trained in computation. In fact, the vast majority of bioinformatics software is now designed for the command line [3]. These individuals must overcome the digital barrier and switch from using a GUI to the command line [4]. As a result, major life-science and medical research institutions are challenged with supporting the analysis of genomic and other large-scale data generated by groups who traditionally have not received computational training [5]. Here we propose a model for addressing this digital barrier in contemporary biology. Our approach helps life-science and medical researchers transition from using a GUI (e.g., Microsoft Excel) to UNIX command line.

Life-science and medical researchers often lack formal training in the use of the command line. These research teams face several unique challenges and opportunities. One approach is for researchers to delegate large-scale data analyses to bioinformatics cores. Delegating analyses to professional bioinformatics researchers can be beneficial, but enabling the researchers to analyze or participate in the analysis of their generated data may be a more sustainable approach. Given sufficient training, researchers should ideally be able to

Box 1. Advantages and Challenges of the Command Line in Bioinformatics

Long-term Reproducibility

The command line has existed since the 1970s and will likely endure, offering long-term reproducibility and sustainability of analytic methods in the scientific community.

Scalability

The command line allows users to implement a large, growing number of discrete tasks with a single action. By contrast, a GUI handles a small number of tasks and has limited scalability.

Control over Data

The command line allows users greater control over raw data. For example, the user can easily parse and transform multi-million-line files, which are often too large to parse and edit using a GUI.

Control over Software

UNIX users can install and use any new software tool, which typically includes many features. In a GUI, users cannot add their own tools to a server and often have limited versions of compatible software that include fewer features.

Communication with the Computer

Analysis of large-scale data relies on commands that must be expressed in a nuanced manner. Text, used in the command line, is the most flexible way of communicating with a computer. GUIs normally lack this functionality.

Simple Design

The command line offers a straightforward, simple design and offers numerous simple, well-designed commands. These simple commands can be easily assembled in the UNIX pipeline to accomplish complex tasks.

The Level of Abstraction

Compared to the command line, GUIs offer a higher level of abstraction. While high abstraction can increase user productivity, a GUI may encourage the researcher to adopt a simplistic 'click a button and get an answer' perspective on bioinformatics analysis.

User-friendliness

The command line is visually less intuitive and requires more memorization for effective operation. First-time users of the command line may have more difficulty executing simple tasks. With GUIs, researchers with limited computational training can accomplish sophisticated computational tasks.

explore additional aspects of data or perform different analyses from what was originally planned. Increasingly, organizers of bioinformatics cores facilities recommend training scientists to at least partially analyze their own data in order to increase the flexibility and scalability of bioinformatics core units [6,7].

Another approach is to develop a GUI, such as Galaxy, that allows researchers with limited computational background to easily create, run, and troubleshoot analytical pipelines [8]. While useful in many cases, providing researchers an alternative interface for command-line tools has several drawbacks. These interfaces are more limited in computational power, and the GUI inherently limits the researcher's flexibility in analyzing data. In addition, a user with no prior experience will most likely use default settings that may or may not be relevant to their biological problem of interest [9]. Data analysis requires a command to be expressed in a finely detailed and nuanced manner, which Galaxy (or any other GUI) cannot support. In general, text is a more flexible method of communication compared to populating forms and clicking on buttons used in a GUI. While useful in many cases, GUIs may encourage the researcher to adopt a 'click a button and get an answer' perspective on bioinformatics.

To overcome these limitations, we believe that life-science and medical research groups should receive training and resources to analyze the data that they generate. This 'training and collaboration' model encourages research groups to efficiently complete projects and advance their own skills. However, researchers lacking a background in computer science are often intimidated by systems that lack a graphical interface and require inputting code, such as UNIX.

Traditional educational models for life-science and medical researchers at the undergraduate and graduate level do not include computational training [10]. Learning command-line tools as an advanced scholar is challenging, because university computer science courses are part of intensive, multiyear curricula. Introductory-level computer science courses build the learners' background knowledge, are time-consuming, and are inflexibly scheduled during the academic year. Therefore, there is growing demand for bioinformatics training in data or statistical analysis and interpretation skills, particularly in the format of dedicated small-group workshops led by skilled trainers [11]. Several online initiatives are available for introducing the command line to novice users.[i] In addition, we maintain a freely available online catalogue[i] of resources and published papers, which includes a feature for submitting newly developed resources.

Under this framework, we developed a 3-day series of workshops that train students with no prior computational background to use the command line for analytical tasks (Figure 1). The goal of these workshops is for life-science and medical researchers to acquire just enough knowledge and skills to independently use small, yet powerful, commands for rapid exploration and modification of data. While many workshops charge a fee and restrict accessibility of materials[i], we freely distribute materials and video recordings so people can take the workshop online. Here, we describe how to replicate our workshop at any given institution.

Over a span of 6 years, postdoctoral scholars affiliated with the Collaboratory of the Institute for Quantitative and Computational Biosciences (QCBio) at the University of California, Los Angeles (UCLA) have taught these workshops to over 400 people. We engage undergraduates, graduate students, postdoctoral scholars, and faculty from biological and medical disciplines in groups of approximately 15–20 individuals.

Qualitative feedback has been overwhelmingly positive; after nine hours of instruction, participants report that they are able to effectively use the command line to manage and analyze their data.

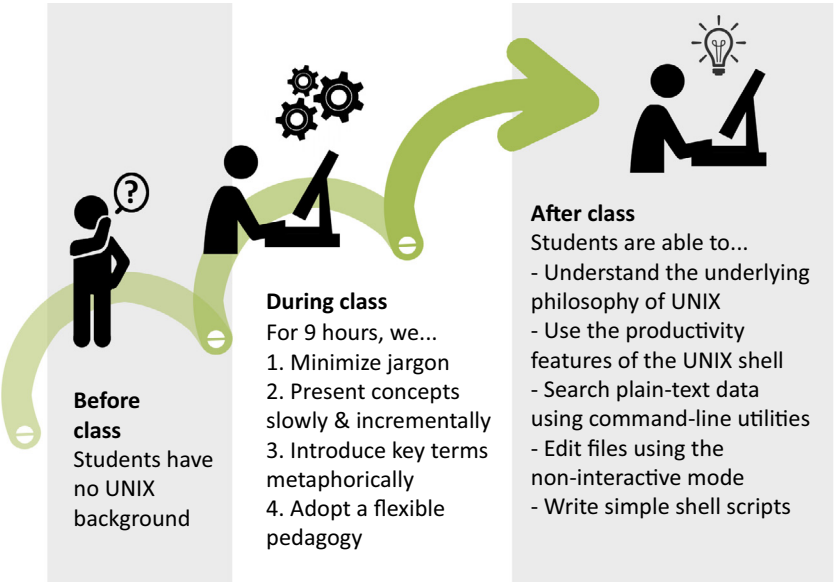

Figure 1. Workshops Provide First-time UNIX Users with an Incremental Introduction to the Skills Necessary for Completing Analytical Tasks with the Command Line.

Many report mastering fundamental skills, such as directly entering commands line-by-line into a terminal – without the familiar aid of a GUI.

We offer four specific suggestions for teaching the UNIX command line. First, we minimize jargon and discipline-specific technical terminology. When unavoidable, we introduce terminology with a clear definition and explanation of the term's context and application. Second, we present concepts at a slow and incremental pace. First-time learners often advance in the workshop at different paces. We regularly pause the course to walk around the class and provide one-on-one tutoring. These individual sessions provide an opportunity for students to ask questions that they might otherwise not ask in front of the class. In our workshops, new concepts are introduced stepwise, building upon the previous concept. Thus, we view such interruptions as a way to guarantee that students acquire the necessary skills before moving on to the next unit.

Third, we simplify the technical terminology by introducing key terms metaphorically, an approach used in other workshops[ii]. Rather than cultivating a deep understanding of fundamental computer science principles, we encourage learners in our workshops to quickly assimilate introduced techniques and apply skills within the context of their research project. For example, the concept of a 'variable' in computer science is highly technical and requires substantial knowledge of informatics in order to grasp. We introduce this term metaphorically; as in, 'the variable is a box where you store the numbers.'

Finally, we adopt a flexible pedagogy. We follow no set educational philosophy; instead, we introduce fundamental concepts as needed and offer a substantial number of hand-on examples and personal guidance to consolidate the learner's newly acquired knowledge.

We also developed an evaluation plan to continually improve the quality of our workshops. Before and after each workshop, we administer a quiz to assess the efficiency of the training model. Observational results suggest that our model can successfully train first-time users of command-line input systems to complete data analysis tasks. In one group, we observe complete elimination of the 'little knowledge' category – all first-time learners moved from 0–70% to 25–100% after a short series of intensive workshops totaling nine instructional hours. We also see a shift of the entire cohort from including scores of 0–100% to scores of 25–100%.

Training life-science and medical researchers in using UNIX to manage and analyze data appears to be successful with a series of workshops. We believe that any institution can replicate our approach of unpacking these computational skills in an approachable and digestible manner.

Our approach is easily reproduced and particularly useful for institutions where researchers from the life sciences and medical sciences engage in big-data projects and frequently outsource computational analysis. An ability to analyze high-throughput data represents a competitive advantage for life-science and medical researchers in today's age of big data and next generation sequencing. UNIX is an 'entry ticket' to bioinformatics; by gaining familiarity with UNIX, researchers may find it easier to engage with other applications and programming languages that are commonly used in computational biology.

We have also developed other workshops (n = 15) that use similar teaching strategies. These workshops include, among others, 'Intro to R and Bioconductor' and 'Informatics for RNA-sequence Analysis.' Workshop materials of all workshops conducted through QCBio are publically available[iii].

Acknowledgments

We thank Jessica Jimenez for administering the surveys in the workshops and analyzing the survey results.

Resources

[i]https://smangul1.github.io/command-line-teaching/
[ii]https://smangul1.github.io/command-line-teaching/
[iii]https://qcb.ucla.edu/collaboratory/workshops/

[1]Department of Computer Science, University of California Los Angeles, 580 Portola Plaza, Los Angeles, CA 90095, USA
[2]Institute for Quantitative and Computational Biosciences, Boyer Hall, 611 Charles Young Drive, UCLA, Los Angeles, CA 90095, USA
[3]Department of Microbiology, Immunology and Molecular Genetics, University of California Los Angeles, 611 Charles E Young Drive East, Los Angeles, CA 90095, USA
[4]Department of Molecular, Cell and Developmental Biology, University of California Los Angeles, 801 Hilgard Avenue, Los Angeles, CA 90095, USA
[5]Department of Human Genetics, University of California Los Angeles, 695 Charles E. Young Drive South, Los Angeles, CA 90095, USA
[†]These authors contributed equally to the paper.

*Correspondence: smangul@ucla.edu (S. Mangul).
http://dx.doi.org/10.1016/j.tibtech.2017.06.007

References

1. Markowetz, F. (2017) All biology is computational biology. *PLoS Biol.* 15, 2002050
2. Stevens, H. (2013) *Life Out of Sequence: A Data-Driven History of Bioinformatics,* University of Chicago Press
3. Altschul, S. *et al.* (2013) The anatomy of successful computational biology software. *Nat. Biotechnol.* 31, 894
4. Price, M. (2012) Computational biologists: the next pharma scientists. *Sci. Careers* Published online April 13, 2012. http://dx.doi.org/10.1126/science.caredit.a1200041
5. Miller, L.A. and Alben, S. (2012) Interfacing mathematics and biology: a discussion on training, research, collaboration, and funding. *Integr. Comp. Biol.* 52, 616–621
6. MacLean, D. and Kamoun, S.K. (2012) Big data in small places. *Nat. Biotechnol.* 30, 33–34
7. Kallioniemi, O. *et al.* (2011) On the organization of bioinformatics core services in biology-based research institutes. *Bioinformatics* 27, 1345
8. Weber, R.J. *et al.* (2017) Computational tools and workflows in metabolomics: An international survey highlights the opportunity for harmonisation through Galaxy. *Metabolomics* 13, 12
9. Schatz, M.C. (2010) The missing graphical user interface for genomics. *Genome Biol.* 11, 128
10. Feser, J. *et al.* (2013) On the edge of mathematics and biology integration: improving quantitative skills in undergraduate biology education. *CBE Life Sci. Educ.* 12, 124–128
11. Barone, L. *et al.* (2017) Unmet needs for analyzing biological big data: a survey of 704 NSF principal investigators. *bioRxiv* http://dx.doi.org/10.1101/108555 Posted online February 14, 2017