Expanding Undergraduate Exposure to Computer Science Subfields: Resources and Lessons from a Hands-on Computational Biology Workshop

Layla Oesper
loesper@carleton.edu
Carleton College
Northfield, Minnesota, USA

Anya Vostinar
vostinar@grinnell.edu
Grinnell College
Grinnell, Iowa, USA

ABSTRACT

Computational biology is an exciting and ever-widening interdisciplinary field. Expanding the participation of undergraduate students in this field will help to inspire and train the next generation of scientists necessary to support this growing area. However, students at smaller institutions, such as those focused on undergraduate education, may not have access to courses related to or even faculty interested in computational biology. Providing more opportunities for such undergraduate students to be exposed to computational biology, or other subfields within computer science, will be important for ensuring these students are included in the pipeline of scientists contributing to these diverse fields. To this end, we hosted a computational biology workshop that brought together undergraduate students from three different liberal arts colleges. The goal of the workshop was to provide an introduction to how computer science can be used to help answer important problems in biology. A diverse set of six faculty members from different institutions each created and taught a hands-on module as an introduction to a different area of computational biology at the workshop. We describe how we went about organizing this undergraduate workshop, summarize the workshop materials that are freely available, and discuss the outcomes and lessons learned from the workshop. We further propose that the workshop structure used is adaptable to other subfields of computer science. Workshop materials available at the workshop website: https://sites.google.com/carleton.edu/compbioworkshop2018/home.

CCS CONCEPTS

• **Social and professional topics** → **Computing education**; • **Applied computing** → **Computational biology**.

KEYWORDS

computational biology, undergraduate education, workshop

ACM Reference Format:
Layla Oesper and Anya Vostinar. 2020. Expanding Undergraduate Exposure to Computer Science Subfields: Resources and Lessons from a Hands-on Computational Biology Workshop. In *The 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20), March 11–14, 2020, Portland, OR, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3328778.3366909

SIGCSE '20, March 11–14, 2020, Portland, OR, USA

ACM ISBN 978-1-4503-6793-6/20/03.
https://doi.org/10.1145/3328778.3366909

1 INTRODUCTION

In recent years biology has become increasingly intertwined with computer science, leading to the birth of new fields of study known as computational biology or bioinformatics [15]. The size and complexity of modern day biological datasets now require novel computational methods to analyze, store, and query [16]. In addition, ever more complex computational models are being created to gain insight into these biological systems [4]. Thus, there is an increasing need for scientists with computational skills and an interest in problems in biology [23, 27].

Concurrently, there has been an expansion in computational biology educational programs [19] and curricular efforts [8, 25] aimed specifically at training undergraduates. However, the lack of educators trained in the field of computational biology or bioinformatics, specifically at liberal arts colleges and regional universities [28], means that undergraduate students at these types of institutions are less likely to have exposure to this growing field.

While some institutions have experimented with introductory computer science courses that focus on problems in biology [5, 11], most smaller institutions do not have the resources or expertise to add such new courses to their curriculum. Furthermore, many of the resources that do exist for training undergraduates in computational biology are aimed at life-science students [7, 14, 26] rather than computer science students. However, the nearly universal boom in students studying computer science [29] means that now is an opportune moment to provide more resources aimed at introducing undergraduate students with a background in computer science to how their skills can be applied to important problems in biology.

Short courses, or workshops have been used previously to provide bioinformatics training. For example, a collaboration between the European Molecular Biology Laboratory-European Bioinformatics Institute (EMBL-EBI) and 2 Australian organizations has led to a series of successful workshops at training bioinformatics educators [17]. Other programs such as the Computational Genomics Summer Institute (CGSI) [1] are aimed at providing more advanced training to graduate students or post-docs. But, little has been done with regards to using short courses or workshops to expose undergraduate students to computational biology or bioinformatics.

This experience report describes a hands-on undergraduate computational biology workshop we organized in the fall of 2018. The workshop introduced students at small liberal arts colleges with a

background in computer science to how computation can be applied to important problems in biology. The main contributions of this work to the education community are the specific workshop materials, available on the workshop website, and our descriptions of how to successfully organize a undergraduate workshop aimed at exposing students from undergraduate only institutions to a broader range of computer science subfields than they experience at their home institution. The structure of the workshop can be applied to any other subfield of computer science where students are typically unable to get a sense of the scope of the subfield at small institutions. Specifically, in section 2 we provide an overview of the structure of the workshop, the faculty who contributed to the workshop content and the students who attended. In section 3 we provide an overview of the content taught at the workshop. All materials associated with the workshop are freely available to the education community through the workshop website. In section 4 we provide an analysis of the outcomes from the workshop for both students and educators, as well as what aspects of the workshop were successful or could be improved in future offerings. Finally, in section 5 we provide some final reflections on what we learned from organizing this workshop and how we hope these resources will be used by the education community in the future.

2 WORKSHOP DETAILS

We organized the first Undergraduate Computational Biology Workshop to occur over a weekend in September 2018 on the campus of Carleton College in Northfield, MN. In this section we outline details regarding the format and organization of the workshop.

2.1 Workshop Format

The one and a half day workshop consisted of six distinct 90 minute modules, each created and taught by a faculty member whose research lies in a different area of computational biology. Each module contained a background component and a hands-on component. The modules were taught in a large computer lab where all necessary software for each module had been pre-installed on the computers. Instructions for the hands-on components and other related material were available on the workshop website. Students worked in pairs on the lab computers when completing the modules.

In addition to the modules, the workshop included all meals for participants, as well as space and time for socialization between students and module leaders to help facilitate interaction. Out of town participants were provided lodging. The workshop also included a sit-down dinner where students and module leaders could interact outside of the computer lab.

2.2 Module Leaders

Six module leaders, each of whom was a full-time faculty member at a different undergraduate-focused small liberal arts college, taught at the workshop. The group of module leaders was extremely diverse on a number of different axes. The group included four members of computer science departments and two members of biology departments. There were also four women and at least two non-Caucasians in the group.

The participating module leaders were specifically selected to have a broad set of research interests, which are reflected in the set of modules they created for the workshop. The intent for having such a broad spectrum of modules was to enable students to get a wide-lens view of how computation can be applied to problems in biology. An additional benefit is that educators using the workshop modules will have ready resources for expanding the type of computational biology content their students see.

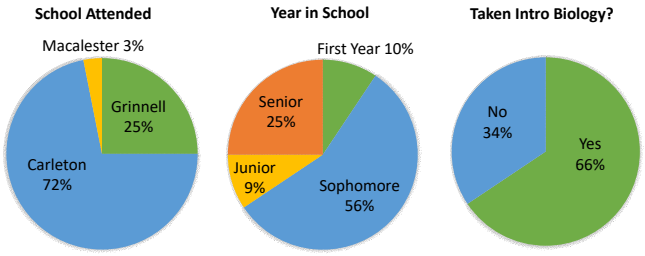

Figure 1: Statistics about the 32 students who registered for the undergraduate computational biology workshop.

Furthermore, since computational biology is such a new and diverse field, researchers in different sub-areas tend to attend separate conferences and have few venues for in-person interaction. For members of small liberal arts colleges, where they are typically the only faculty at their institution working in the field of computational biology, they may have little direct interaction with other faculty working in their area at liberal arts colleges. The workshop provided the module leaders with a novel avenue for interaction with others working in related areas at similar institutions. Many of the module leaders had never met each other before the workshop. Even the workshop coordinator who selected the invited module leaders had never met 1/3 of the module leaders before inviting them to participate. Thus, the workshop doubled as a valuable networking opportunity for the module leaders, as well as an educational experience for the student participants.

2.3 Student Participants

The workshop was attended by 32 undergraduate students from three liberal arts colleges located in the region (Carleton College, Grinnell College and Macalester College). The only pre-requisite for participation in the workshop was that the student had completed the equivalent of a CS1 course. Of the 32 registered student participants, the majority (72%) were from the host institution. Student participants represented all years in school, with the largest component being sophomores (56%) and 34% of all participants had not previously taken an introduction to biology course (see Figure 1). According to pre-workshop surveys, completed by 22 of the 32 students, student participants covered a wide-variety of majors or intended majors (41% of respondents indicated two majors) and had relatively little previous experience with computational biology (see Figure 2).

Students at the host institution were recruited through emails to computer science interest lists and targeted emails to specific biology and computer science class lists. The workshop organizer also contacted faculty members, including some module leaders, at surrounding liberal arts colleges who forwarded the call for participants to students at their own institutions.

2.4 Student Assistants

Six students at the host institution were hired to help with the coordination of the workshop. Their main responsibility was to beta-test the modules in the computer lab where the workshop would be held. Each module was beta-tested by two students, with different levels of computer science experience when possible. These students ran through each module and noted points of confusion, questions that arose while completing the module, and any technical issues they encountered. The feedback from these students was then forwarded to the module leaders who then updated their module. In the case of technical issues, the comments were forwarded to the technical staff in charge of installing the correct software in the computer lab for remediation.

Each student assistant also completed a 3-hour shift at the workshop to help with setup, breakdown, and any other issues that arose. Many students were able to attend the majority of the workshop during their shift.

3 MODULE CONTENT

In this section we outline the broad structure of each of the modules and briefly describe the contents of each one.

3.1 Module Format

Each module introduces students to a different area of how computation can be applied to biology. Most module leaders created a module that serves as a broad introduction to their research area. Each module (available on the workshop website) contains the following sections:

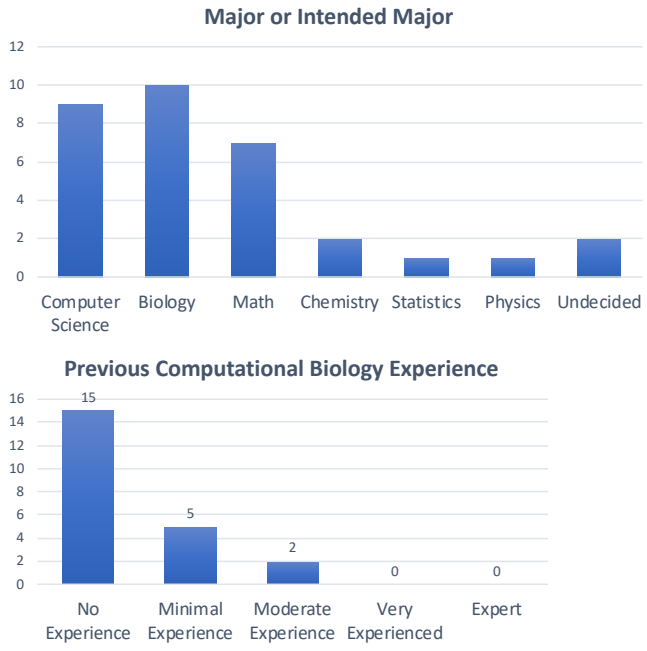

Figure 2: Results from pre-workshop survey completed by 22 of the 32 registered participants. The majority of participants had no previous computational biology experience.

- **Summary** - A brief description of the topic to be covered in the module and what students will learn by completing the module.
- **Presentation Materials** - Any materials (e.g., slides) used during the background section of the module.
- **Hands-on Exercises** - Specific instructions for the hands-on component of the module.
- **Associated Materials/Files** - Many of the modules analyze specific datasets or have cheatsheets for tools used as part of the module.
- **Program/Software Requirements** - A complete list of the necessary software is included for each module so that they can be re-used at a different location.
- **Advanced Material** - Additional information, such as related research papers, for students interested in learning more about the topic covered in the module.
- **Instructor Notes** - Additional details or information for instructors choosing to use the module.

3.2 Module 1: Finding Friends in Molecular Interaction Networks

3.2.1 Summary. Cells respond to external signals through protein-protein interactions. These interactions are often represented as a graph, and algorithms from graph theory can be used to generate hypotheses about protein regulation. This module introduces the computational problem of identifying candidate regulators of a specific protein of interest using molecular interaction networks. In the hands-on component, students predict novel regulators of *Fog signaling*, which is involved in changing the shape of a cell. Specifically, students try their hand at identifying candidate regulators in a newly-established Drosophila interaction network, and visualize their results using graph visualization software.

3.2.2 Hands-On Details. The hands-on component of this module uses the online coding environment Repl.it [2] to provide students with starter code and, therefore, does not require anything other than a browser on the user's computer. The exercise is broken into two separate components. In the first component students use Python3 to explore working with graphs using the NetworkX package [10]. Students modify the existing code to compute basic graph statistics such as counting the number of edges or nodes in an existing graph. In the second component, students explore ideas to rank unlabeled nodes in a given graph and are provided code to use the GraphSpace [6] tool to visualize their ranked nodes. In particular, students get the chance to implement and try out their ideas on a fly interaction network. Figure 3 shows two of the visualizations created by students during the workshop.

3.3 Module 2: Peptide and Protein Identification using MS/MS Data on the Galaxy-P platform

3.3.1 Summary. Bottom-up proteomics is a technique for studying proteins that involves digesting proteins to fragment peptides prior to identification using mass spectrometry. Masses of digested peptides are compared with those predicted from a sequence database to identify peptides. Proteins are then identified by analyzing

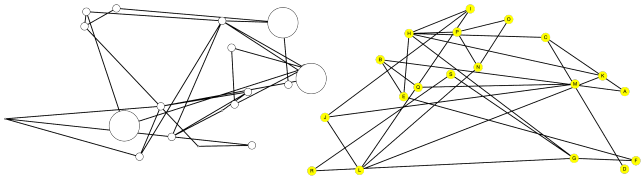

Figure 3: Two of the graph visualizations created by students during the workshop.

peptides present in the sample. This module is an introduction to peptide and protein identification using MS/MS data on the Galaxy-P platform [22]. Galaxy-P is an extension of Galaxy, a platform developed to enable reproducible data-intensive research by providing complex computational software that often requires advanced computer skills via a web-interface.

3.3.2 Hands-On Details. This component enables students to learn about and use the Galaxy-P platform. Galaxy-P is a multi-omics informatics platform focusing on integrative analysis of genomics, transcriptomics, mass spectrometry-based proteomics, and metabolomics data. During the workshop a special cloud instance of Galaxy-P was used by workshop participants, but subsequent users can download and install their own instance of the tool. The module walks students through the basics the Galaxy framework and of peptide and protein identification using MS/MS data. Students are also introduced to a companion tool, Peptide Shaker [24] that uses output they create via the Galaxy-P platform to organize and analyze peptide spectrum matches (PSMs).

3.4 Module 3: Identifying Somatic Mutations in mtDNA

3.4.1 Summary. Exploration of variation in DNA sequences in humans is a key element towards understanding many diseases, especially cancer. In this module students learn about comparing paired tumor and normal samples to find somatic (non-inherited) mutations from DNA sequencing data of cancer patients. Specifically, students are introduced to the basics of DNA sequencing data and how computational approaches can be applied to such data to identify genomic variants in comparison to a reference genome. Students are also introduced to how mitochondrial dysfunction may play a role in different human diseases.

3.4.2 Hands-On Details. This module walks students through the steps to identify somatic mutations in mitochondrial DNA from DNA sequencing data using both written questions and guided use of several software tools. Specifically, students first learn some basic facts about the BAM (Binary Alignment Mapping) file formats used to represent sequencing data. Students then get practice at manipulating and "viewing" BAM files using the command-line tool called Samtools [13]. Students then explore how Samtools, when combined with another tool called VarScan2 [12], can be used to identify genomic variations. Finally, students use these tools to find somatic mutations in mitochondrial DNA (mtDNA).

3.5 Module 4: Machine Learning for Biological Data

3.5.1 Summary. Over the past few decades, biology has undergone a revolution of data as powerful high-throughput experiments continue to decrease in cost and increase in scope. This "deluge of data", however, is only as useful as the quantitative analysis methods available to make sense of the information. Scientists have increasingly looked towards the field of artificial intelligence, in particular, the subarea of machine learning, for solutions to these data analysis deficiencies. In this module, students are introduced the basic frameworks of machine learning (supervised and unsupervised learning) and apply them to biological problems. The module first looks at historical applications - the use of gene expression data to uncover gene function and to provide predictive markers for disease. Then, the module provides a survey of recent trends in computational biology including sequence analysis and structure prediction.

3.5.2 Hands-On Details. This component contains two tasks that can be completed in any order. The first task walks students through clustering gene-expression data for a yeast dataset. The second task has students train a model for analyzing gene expression data of colon cancer patients. In both parts, students run and modify existing Python3 code. All starter code, instructions and data are included in a Git repository that students can check out. The provided code makes use of the Numpy [20] and Scikit-Learn [21] Python modules and includes links to cheat sheets for these modules.

3.6 Module 5: Binning Genomes from Metagenomes

3.6.1 Summary. An exciting bioinformatics advance from the past few years has been our ability to disentangle microbial genomes from environmental samples called metagenomes. This has allowed scientists to study the full diversity of life without needing to culture these microbes - and the majority of microbes have not yet been cultivated. Our ability to "bin", or recover, genomes from metagenomes has revolutionized our view of the tree of life, and changed our understanding of our own place within that tree of life. In this module students learn how to bin genomes from metagenomes.

3.6.2 Hands-On Details. This module walks students through the use of command line tools for analyzing metagenomics data. Specifically, the module focuses on the use of multi-purpose tool Anvi'o [9]. Students walk through step-by-step instructions to download metagenomic sequencing data and apply common analysis techniques such as assembly and mapping to eventually produce a set of "bins", or reconstructed genomes. The module also provides students with a visualization tool for exploring and analyzing their bins. Other advanced analysis options are also available within the module.

3.7 Module 6: Artificial Life Agent-Based Simulations using the GPU

3.7.1 Summary. Artificial life is a field that combines computing and biology in a different way than bioinformatics. Using object-oriented programming, we can code the behavior for individual agents and then fill a world with instances of those agents. In this

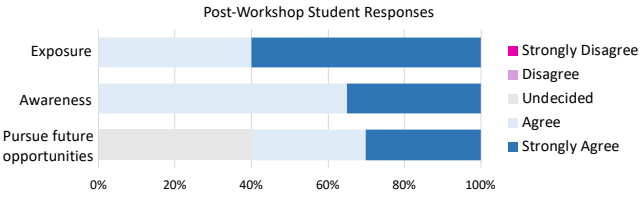

Figure 4: Student post-workshop surveys indicate that they had increased exposure and awareness of how computation can be applied to problems in biology. To a slightly lesser degree students indicated that they intended to continue to pursue opportunities related to computational biology.

module students fill a world with predator and prey agents. The predator agents are programmed to chase and eat the prey, and the prey are programmed to try to escape the predators. By observing these simple individual-level behaviors, students will see what the population as a whole does.

3.7.2 Hands-On Details. This component requires that a user have a CUDA GPU from NVIDIA. If a user does not have this hardware and still wishes to run the module, they will need to use Amazon Web Services or another service that allows a user to run with CUDA GPU. In this module students explore how changing different parameters affect the resulting predator and prey populations. In particular, students will aim to identify Lotka-Volterra oscillations in their simulations.

4 OUTCOMES AND LESSONS LEARNED

We used post-workshop surveys to assess the impact of the workshop on both students and module leaders. We also performed a rigorous analysis of what aspects of the workshop went well and how future offerings could be improved.

4.1 Student Outcomes

We assessed student response to the workshop using a survey where students were asked to use a Likert scale to rate their level of agreement with several statements about their experience with the event. Figure 4 shows the distribution of the 20 student responses to the following three statements: **(1) Exposure:** The workshop exposed me to novel and exciting applications involving computation and biology; **(2) Awareness:** I am aware of a variety of ways that computation can be combined with biology; and **(3) Pursue future opportunities:** I am likely to pursue future opportunities (REUs, jobs, graduate school, etc.) that are related to computational biology. Overall, students showed a high level of agreement with these statements, indicating that the workshop achieved its stated goal of giving students a hands-on introduction to how computer science can be used to help answer important problems in biology.

4.2 Module Leader Outcomes

We assessed module leader response to the workshop using a survey where they were asked to use a Likert scale to rate their level of agreement with several statements about their experience with the event. Most of the module leaders indicated that they either agreed (2 of 6) or strongly agreed (2 of 6) that they would use a module created by another module leader in their own future classes or outreach. Furthermore, all module leaders agreed that the workshop had helped to expand their computational biology professional network. Thus, we found that the workshop had a positive impact on the professional careers of the module leaders.

4.3 Lessons Learned

After the workshop we used both student and module leader comments from surveys as well as observations during the workshop to identify aspects of the workshop that went well and areas that could be improved for any potential future offerings.

4.3.1 Aspects of the workshop that went well. The use of student assistants prior to the workshop for beta-testing the modules was extremely useful. In addition to catching technical issues, they were also able to identify potential areas of confusion with the modules, especially for students with limited computer science background. As a result, the modules were much more robust. Furthermore, we found that the students who served as student assistants represented an even more broad swath of students (including non-STEM majors) who were then exposed to the module material.

Since the workshop occurred during the school year, we were cognizant that the student participants would need to balance classwork with their attendance at the workshop. We found that the one and one half day length of the workshop (with generous time for breaks) provided a good balance by allowing time for a wide array of different module topics, but also allowed students time to dedicate to their studies. The half day on the second day of the workshop also allowed time for students from other colleges to travel back to their home institutions. Finally, the length of time encouraged students to interact with each other, including across college boundaries.

4.3.2 Aspects of the workshop that could be improved. We identified three areas of the workshop that could be improved in future offerings.

First, since we only required that students have the equivalent of a CS1 course, the participating students had a wide range of computational experience from students who had only taken one CS course, to seniors that had completed the major. This made it challenging to have modules pitched at the correct level for all student participants. In future offerings of the workshop, we will try different approaches to better match module content to students' computational background. Specifically, we plan to either break modules into levels (e.g., beginner, intermediate, advanced) or to have each module contain different hands-on components for students with different levels of background. This approach will allow us to still keep the workshop open to as many students as possible.

Second, the module materials were only discoverable to those who found the associated website either through a workshop organizer, a module leader, advertising materials for the workshop, or through tweets about the event. We plan to address this issue by partnering with QUBES [3], an organization specifically dedicated to addressing challenges in quantitative biology education. In particular, QUBES offers an online infrastructure for hosting workshops and the associated materials that will automatically make these

materials more easily accessible to the wider quantitative biology community. We will create a group on the QUBES platform to host all materials from previous and future offerings of the workshop.

Finally, several of the modules required specialty software (and in one case hardware, i.e., NVIDIA GPU) to be installed prior to the workshop. This can make it more challenging for others to adopt the use of these modules or for students to try out different modules on their own. In future offerings of the workshop, we hope to move some modules to online coding environments, such as Repl.it, that only require a user to have a web browser. Other options would be to use other container style environments such as a Docker image [18] or the online workspaces provided by QUBES [3] that allow participants access to an online environment with associated software already installed, rather than having to install locally.

5 CONCLUSIONS

We found that the 2018 Undergraduate Computational Biology Workshop was a success towards our goal of exposing computer science students to how computation can be applied to important problems in biology. However, more work is needed to ensure that more undergraduate students have the opportunity to be exposed to computational biology, especially those at small liberal arts colleges where no current faculty have experience in the field. We hope that by making all the resources from this workshop available to the larger education community, other educators will be able to utilize these resources in their own classes. We also expect that the lessons learned from the initial offering of this workshop will be useful towards improved iterations of the workshop in future years that will enable more undergraduates at liberal arts colleges to be exposed to the exciting and growing world of computational biology. Finally, we recommend the structure used in this workshop to enable undergraduates to experience the breadth of applications of any computer science subfield.

6 ACKNOWLEDGMENTS

We thank the other module leaders, Rika Anderson, Getiria Onsongo, Anna Ritz, Ameet Soni and Catie Welsh, for creating the content for the modules described here. This workshop was funded by the National Science Foundation CRII award (IIS-1657380).

REFERENCES

[1] 2019. http://computationalgenomics.bioinformatics.ucla.edu/
[2] 2019. https://repl.it/repls
[3] 2019. https://qubeshub.org/
[4] Mark A Bedau. 2003. Artificial life: organization, adaptation and complexity from the bottom up. *Trends in cognitive sciences* 7, 11 (2003), 505–512.
[5] Tanya Berger-Wolf, Boris Igic, Cynthia Taylor, Robert Sloan, and Rachel Poretsky. 2018. A Biology-themed Introductory CS Course at a Large, Diverse Public University. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. ACM, 233–238.
[6] Aditya Bharadwaj, Divit P Singh, Anna Ritz, Allison N Tegge, Christopher L Poirel, Pavel Kraikivski, Neil Adames, Kurt Luther, Shiv D Kale, Jean Peccoud, John J Tyson, and T M Murali. 2017. GraphSpace: stimulating interdisciplinary collaborations in network biology. *Bioinformatics* 33, 19 (Oct 2017), 3134–3136. https://doi.org/10.1093/bioinformatics/btx382
[7] Maureen A Carey and Jason A Papin. 2018. Ten simple rules for biologists learning to program. *PLoS Comput Biol* 14, 1 (01 2018), e1005871. https://doi.org/10.1371/journal.pcbi.1005871
[8] Travis Doom, Michael Raymer, Dan Krane, and Oscar Garcia. 2002. A proposed undergraduate bioinformatics curriculum for computer scientists. In *ACM SIGCSE Bulletin*, Vol. 34. ACM, 78–81.
[9] A Murat Eren, Özcan C Esen, Christopher Quince, Joseph H Vineis, Hilary G Morrison, Mitchell L Sogin, and Tom O Delmont. 2015. Anvi'o: an advanced analysis and visualization platform for 'omics data. *PeerJ* 3 (2015), e1319.
[10] Aric Hagberg, Pieter Swart, and Daniel S Chult. 2008. *Exploring network structure, dynamics, and function using NetworkX*. Technical Report. Los Alamos National Lab.(LANL), Los Alamos, NM (United States).
[11] Maria Klawe. 2013. Increasing female participation in computing: The Harvey Mudd College story. *Computer* 46, 3 (2013), 56–58.
[12] Daniel C Koboldt, Qunyuan Zhang, David E Larson, Dong Shen, Michael D McLellan, Ling Lin, Christopher A Miller, Elaine R Mardis, Li Ding, and Richard K Wilson. 2012. VarScan 2: somatic mutation and copy number alteration discovery in cancer by exome sequencing. *Genome Res* 22, 3 (Mar 2012), 568–76. https://doi.org/10.1101/gr.129684.111
[13] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, Richard Durbin, and 1000 Genome Project Data Processing Subgroup. 2009. The Sequence Alignment/Map format and SAMtools. *Bioinformatics* 25, 16 (Aug 2009), 2078–9. https://doi.org/10.1093/bioinformatics/btp352
[14] Andreas Madlung. 2018. Assessing an effective undergraduate module teaching applied bioinformatics to biology students. *PLoS Comput Biol* 14, 1 (01 2018), e1005872. https://doi.org/10.1371/journal.pcbi.1005872
[15] Florian Markowetz. 2017. All biology is computational biology. *PLoS Biol* 15, 3 (03 2017), e2002050. https://doi.org/10.1371/journal.pbio.2002050
[16] Vivien Marx. 2013. Biology: The big challenges of big data. *Nature* 498, 7453 (Jun 2013), 255–60. https://doi.org/10.1038/498255a
[17] Annette McGrath, Katherine Champ, Catherine A Shang, Ellen van Dam, Cath Brooksbank, and Sarah L Morgan. 2019. From trainees to trainers to instructors: Sustainably building a national capacity in bioinformatics training. *PLoS Comput Biol* 15, 6 (Jun 2019), e1006923. https://doi.org/10.1371/journal.pcbi.1006923
[18] Dirk Merkel. 2014. Docker: lightweight linux containers for consistent development and deployment. *Linux Journal* 2014, 239 (2014), 2.
[19] Nicola Mulder, Russell Schwartz, Michelle D Brazas, Cath Brooksbank, Bruno Gaeta, Sarah L Morgan, Mark A Pauley, Anne Rosenwald, Gabriella Rustici, Michael Sierk, Tandy Warnow, and Lonnie Welch. 2018. The development and application of bioinformatics core competencies to improve bioinformatics training and education. *PLoS Comput Biol* 14, 2 (02 2018), e1005772. https://doi.org/10.1371/journal.pcbi.1005772
[20] Travis E Oliphant. 2006. *A guide to NumPy*. Vol. 1. Trelgol Publishing USA.
[21] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research* 12, Oct (2011), 2825–2830.
[22] Gloria M Sheynkman, James E Johnson, Pratik D Jagtap, Michael R Shortreed, Getiria Onsongo, Brian L Frey, Timothy J Griffin, and Lloyd M Smith. 2014. Using Galaxy-P to leverage RNA-Seq for the discovery of novel protein variations. *BMC Genomics* 15 (Aug 2014), 703. https://doi.org/10.1186/1471-2164-15-703
[23] David R Smith. 2018. Bringing bioinformatics to the scientific masses: As the demand for high-level bioinformatics is growing, training students in the field becomes ever more important. *EMBO Rep* 19, 6 (06 2018). https://doi.org/10.15252/embr.201846262
[24] Marc Vaudel, Julia M Burkhart, René P Zahedi, Eystein Oveland, Frode S Berven, Albert Sickmann, Lennart Martens, and Harald Barsnes. 2015. PeptideShaker enables reanalysis of MS-derived proteomics data sets. *Nat Biotechnol* 33, 1 (Jan 2015), 22–4. https://doi.org/10.1038/nbt.3109
[25] Lonnie Welch, Fran Lewitter, Russell Schwartz, Cath Brooksbank, Predrag Radivojac, Bruno Gaeta, and Maria Victoria Schneider. 2014. Bioinformatics curriculum guidelines: toward a definition of core competencies. *PLOS computational biology* 10, 3 (2014), e1003496.
[26] Melissa A Wilson Sayres, Charles Hauser, Michael Sierk, Srebrenka Robic, Anne G Rosenwald, Todd M Smith, Eric W Triplett, Jason J Williams, Elizabeth Dinsdale, William R Morgan, James M Burnette, 3rd, Samuel S Donovan, Jennifer C Drew, Sarah C R Elgin, Edison R Fowlks, Sebastian Galindo-Gonzalez, Anya L Goodman, Nealy F Grandgenett, Carlos C Goller, John R Jungck, Jeffrey D Newman, William Pearson, Elizabeth F Ryder, Rafael Tosado-Acevedo, William Tapprich, Tammy C Tobin, Arlín Toro-Martínez, Lonnie R Welch, Robin Wright, Lindsay Barone, David Ebenbach, Mindy McWilliams, Kimberly C Olney, and Mark A Pauley. 2018. Bioinformatics core competencies for undergraduate life sciences education. *PLoS One* 13, 6 (2018), e0196878. https://doi.org/10.1371/journal.pone.0196878
[27] Itai Yanai and Eva Chmielnicki. 2017. Computational biologists: moving to the driver's seat. *Genome Biol* 18, 1 (11 2017), 223. https://doi.org/10.1186/s13059-017-1357-1
[28] Yingqian Ada Zhan, Charles Gregory Wray, Sandeep Namburi, Spencer T Glantz, Reinhard Laubenbacher, and Jeffrey H Chuang. 2019. Fostering bioinformatics education through skill development of professors: Big Genomic Data Skills Training for Professors. *PLoS Comput Biol* 15, 6 (Jun 2019), e1007026. https://doi.org/10.1371/journal.pcbi.1007026
[29] Stuart Zweben and Betsy Bizot. 2018. 2017 CRA Taulbee Survey. *Computing Research News* 30, 5 (2018), 1–47.