Published online: May 3, 2018

EMBO reports

Science & Society

Bringing bioinformatics to the scientific masses

As the demand for high-level bioinformatics is growing, training students in the field becomes ever more important

David R Smith

Sometimes, it is not until after you have started a new job that you get a sense of the skills your employer was looking for when hiring you. I had been working as an assistant professor of biology for over 6 months when I finally figured out what kind of scientist the department was actually hoping to get: a bioinformatics fix-it-all, a quick cure for their expanding genomic woes, an in-house computational genius willing to lend his powers to every graduate thesis containing a next-generation sequencing (NGS) dataset. Sadly, I may have disappointed some of my colleagues.

Like many evolutionary biologists, my research relies heavily on molecular sequence data, and because of this I am sometimes categorized as a bioinformatician—admittedly, I occasionally market myself as one—when, in fact, I am merely an end user of sophisticated software, pipelines, and programs that genuine bioinformaticians have designed. Of course, I have picked up some computational skills along the way, enough to assemble and analyze the mitochondrial and chloroplast genomes that encompass my research life. But I am a far cry from being able to perform, for example, the in-depth analyses needed for high-quality metagenomics work. Consequently, when a graduate student or colleague knocks on my office door and says, "Hey, Dave. We just did a ton of next-gen on … and we were hoping that you could help analyze the data," I feign a smile and resist the urge to crawl under my desk.

When I started my job and word got around that a "sequencing" person had arrived, my inbox became bloated with emails from students and coworkers asking for help. In most cases, I tried my best to offer sound advice: "Have you tried this software?" "How about applying this measure to your data?" "Maybe you should just do more sequencing." But usually my suggestions came up short, and it quickly became apparent that I was a false prophet and that a real solution required the guidance of a professional bioinformatician. In hindsight, I was lucky for these shortcomings, for had I been able to provide hands-on help, my own research program would have severely suffered. Indeed, being overworked and overcommitted is a constant complaint I hear from my bioinformatics friends at other institutes, particularly those who have a hard time saying no.

Growing demand

Eventually, the offers for me to collaborate dwindled, but the demand for advanced bioinformatics support within my department did not—and we are not alone. I would argue that there is a growing need across the life sciences for experts and software to carry out high-level bioinformatics tasks. It is getting easier, cheaper, and faster to generate huge amounts of sequence information, but analyzing and interpreting this deluge of data is a major challenge for many researchers.

How then do we bring bioinformatics to the scientific masses? The answer is complex, is multifaceted, and likely involves major changes to the way we train students. As genetic data become all the more ubiquitous in research and healthcare, there are already heated debates about the cost, usability, and availability of bioinformatics and its spoils. Those with first-class bioinformatics skills will be sought after, and the companies that can provide practical solutions to our burgeoning big-data needs will cash in. One area that is bound for great expansion and that could become very lucrative is the development of user-friendly sequence analysis programs.

User-friendly bioinformatics

When I became a postdoc, my first project was to assemble and annotate the entire nuclear genome sequence of a green alga. Naively, I thought that the approaches I had previously used to put together and analyze organelle DNAs from green algae could simply be scaled up—think again, young David. I quickly found out that this is not how bioinformatics works. Just because I was familiar with BLAST and could generate a nucleotide alignment did not mean I had the skills to identify and annotate thousands of nuclear-encoded genes, at least not in a timely fashion. Just because I could point and click my way to a complete organelle DNA sequence did not mean I could use my cursor to generate a polished nuclear genome assembly. Soon, I discovered that merely moving, opening, or exporting the raw data for my project exceeded my computational expertise.

What I needed to be able to do was to work efficiently from a command line, and to read, write, and execute my own scripts in addition to editing and adapting those of others. But, like most biologists attempting

Department of Biology, University of Western Ontario, London, ON, Canada. E-mail: dsmit242@uwo.ca
DOI 10.15252/embr.201846262 | EMBO Reports (2018) 19: e46262 | Published online 3 May 2018

bioinformatics, I wanted to work via a graphical user interface—a GUI. I found this predicament incredibly frustrating because, from my perspective, I had a clear understanding of the *in silico* experiments that needed to be done to complete the project—I just could not do them myself. Moreover, my background in molecular evolution meant that I would have little trouble interpreting the results if only I could run these experiments.

> *"I would argue that there is a growing need across the life sciences for experts and software to carry out high-level bioinformatics tasks."*

Alas, I never assembled that algal nuclear genome; a more experienced bioinformatician eventually did. But in the years after my postdoc, user-friendly bioinformatics software has come a long way, arguably far enough that it is now possible to do sophisticated eukaryotic genomics via a GUI [1]. Take, for instance, CLC Genomics Workbench, owned and distributed by Qiagen. This all-in-one, easy-to-use software suite can do everything from *de novo* eukaryotic genome assemblies to variant detection to epigenomic analyses. It is scalable and customizable and can be used to design workflows. The only catch: It is bloody expensive. In late 2013, I bought a single academic license at a 50% discount for more than CAN$6,000, not including the CAN $1,000 annual maintenance, update and support fee and the additional costs for specific plugins, such as Blast2GO. I tested and priced other similar user-friendly NGS software suites, including Strand NGS (Strand Life Sciences Private Ltd.) and NextGENe (SoftGenetics), but they were equally as expensive.

Most scientists and students cannot afford to purchase these software solutions. There are, however, cheaper alternatives, although they tend to be less powerful than the Mercedes-Benz options. Geneious (developed by Biomatters Ltd.) is a comprehensive suite of molecular biology and NGS tools under a single, polished interface, and a one-year non-commercial license costs only US$395—half that if you are a student. I have been using Geneious since my PhD days; but, admittedly, it is not catered for large-scale eukaryotic genome projects, at least not yet.

If you do employ Geneious, you are apparently in good company. The official homepage (www.geneious.com) claims that it "is the world's leading bioinformatics platform used by over 3,000 universities, institutes and companies in more than 100 countries . . . by all of the top 20 universities globally and by 16 of the 20 largest pharmaceutical companies." Nevertheless, Biomatters Ltd. is still a relatively small company and, more importantly, there is currently no clear winner in the battle for delivering user-friendly bioinformatics programs, despite the rising demand.

Although the fees for commercial software might be a deterrent for many researchers, it is worth noting that most companies offer 2- to 4-week complementary trial periods, allowing potential buyers to test the programs on their own data. Some companies, including DNASTAR, which sells the user-friendly software Lasergene [1], allow clients to purchase individual components of their bioinformatics suite (such as a structural biology toolkit) instead of the entire package, which costs thousands of dollars.

> *". . . whatever bioinformatics software one chooses, its efficiency largely depends on the power of the computer on which it is being run."*

There are also a wide range of freely available bioinformatics programs with intuitive GUIs [2], including Artemis [3] and MEGA [4], which has been downloaded by more than a million and a half people. MEGA is great for doing molecular evolutionary analyses, including phylogenetics, but it is not necessarily designed for genomics or NGS work. The open-source software Unipro UGENE, on the other hand, offers an assortment of NGS programs and can be used to design genomics pipelines and workflows [5]. As one might expect, free bioinformatics platforms tend to be less robust and more finicky than their commercial equivalents. But keep in mind that the private software suites, although often containing proprietary programs, usually rely heavily on the very same open-source algorithms, such as Bowtie, Tophat, and Velvet, that are found in the free ones. Notwithstanding, whatever bioinformatics software one chooses, its efficiency largely depends on the power of the computer on which it is being run.

Sizing up a bioinformatics workstation

Bioinformaticians often treat their computers like living, breathing members of the family. A former colleague of mine was so distraught when his laptop, which he named Big Ben, contracted a virus that he canceled a dinner party and took a day off work (fortunately, Ben lived to see another assembly). Such behavior is not surprising when considering that some bioinformaticians pay more for a computer than they would for a new car, and the language that they use to describe their hardware can be as dense as car-speak: CPUs, quad-cores, megs of RAM, triad chassis, liquid cooling, and so on.

> *"Today, a single genome assembly will rarely yield a high-impact paper; it needs to be a 1,000 or 10,000 genomes."*

Like it or not, examining big data typically requires powerful computers and significant financial investments. Computing power is especially important for massive phylogenetic analyses or *de novo* assemblies of giant genomes. Ask a phylogeneticist how work is going and you are likely to get the answer, "Great. The analysis has been going for 6 weeks and it will be *going* for another five." Complicating matters is the fact that molecular sequence datasets are getting bigger and bigger, as are the expectations for the size and scope of the published results. Today, a single genome assembly will rarely yield a high-impact paper; it needs to be a 1,000 or 10,000 genomes. What is more, user-friendly GUI interfaces are more computationally expensive than their barebones, command-line-driven counterparts, which makes it even more challenging for non-experts to keep up. Running CLC Genomics Workbench or Geneious on a 3-year-old MacBook Air is going to feel slow and clunky and is not a sound strategy for tackling complicated genomics studies.

However, when used on a high-powered, custom-built workstation or server, the same software feels as fast and responsive as a race car.

My laboratory computer, which runs on Linux, has a 10-core Intel Xeon E5 processor, a 4 TB hard drive, a 1 TB solid-state drive, and 384 GB of DDR4 RAM. I do not even know where the power switch is found on this machine—this is the domain of a computer-savvy graduate student—and despite its twelve-thousand-dollar price tag, its size and power pale in comparison with the central computers at major research laboratories and genome centers. I also have access to a supercomputing network at my university, called SHARCNET, comprising ~ 40,000 processors and 20 PB of storage. This is a great asset, specifically to bioinformaticians and other big-data scientists, but accessing and using SHARCNET is not straightforward, which has been my experience with supercomputers in general. For instance, interacting with SHARCNET, including logging on and running and installing programs, is almost entirely limited to a command line and requires a good understanding of the underlying language and scripts, not to mention that there can be long lineups and wait times before you can start an analysis. Some of my coworkers are keen to use SHARCNET, but do not have the computational wherewithal nor the time and energy needed to acquire it. I imagine that the same problem exists on other campuses with supercomputers.

"If scientists lose sight of this fact and blindly embrace technological streamlining and outsourcing, they take the risk of becoming data-generating, grant-writing machines."

The good news is that personal computers are becoming so powerful that it might not be long before the average scientist or student will be able to use a laptop, tablet, or smartphone to examine very large datasets, such as an entire eukaryotic genome, provided they have access to user-friendly software. There is also the strong possibility that in the near future, many of us will not even be analyzing our own bioinformatics data, but instead will outsource the task to someone else.

Subcontracting bioinformatics

Outsourcing bioinformatics experiments might sound like a troubling proposition to some, but it is not a new idea. In fact, it is precisely what computationally challenged researchers have been doing for decades by outsourcing their analyses to expert collaborators, postdocs, technicians, or students. However, subcontracting has shifted away from academic collaboration and recruitment toward a model in which private companies carry out bioinformatics, sometimes at great cost to the researcher.

Over the past 5 years, I have participated in various protist genome projects, and together, my collaborators and I have spent tens of thousands of dollars on commercial sequencing. We also employed some of these same sequencing companies to perform downstream bioinformatics analyses, including *de novo* assemblies, hybrid assemblies, whole-genome annotations, and structural predictions. The outsourcing of these *in silico* studies was expensive—about CAN$2,000 for a draft genome with a basic annotation—but it saved us a lot of time and meant that we could start exploring the data immediately. And because we were spending so much money on NGS, it was easy to justify spending a little more on fast, professional bioinformatics work. The companies know this and will try to profit as much as possible: "Dear, Prof. Smith. Are you sure you don't want to add a reference genome alignment to your Illumina sequencing run? We are currently offering a 25% discount for the month of December."

A day rarely goes by without spam from businesses promoting their bioinformatics products, usually with aggravating slogans like "Think SMRT and learn more about our state-of-the-art genome assembly pipeline by clicking here." "RNA-Seq got you down? Then let our certified bioinformaticians help you find novel transcripts, differential expressions, and functional annotations." Or even more boastful: "We can meet ALL your bioinformatics needs." My own experience with commercial bioinformatics has been positive, but I have treated it more as a means for fast preliminary analyses rather than a route to publication-ready results. I usually end up revising or redoing many of the experiments I initially paid for—and once I have paid, it can be very hard to get the companies to correct any mistakes they made in their work. As the market for commercial bioinformatics expands and becomes more competitive, the consumer will surely be presented with more options, greater personalization and customization, and hopefully higher quality results.

"The idea that coding is a crucial asset in today's workforce and one that can help students develop strong problem-solving capabilities has spurred various educational programs..."

One of the consequences of such widespread outsourcing and commercialization is that scientists could become less like scholars and more like CEOs acquiring and managing grants. Another danger from the farming out of bioinformatics is that investigators will lose touch with the theories and techniques used to generate their data and results, not to mention the implicit complications of having these data (often generated using public money) in the hands of private companies. As the history of science has proven time and again, major advancements come from years of sustained, hands-on involvement with the experiments. If scientists lose sight of this fact and blindly embrace technological streamlining and outsourcing, they take the risk of becoming data-generating, grant-writing machines. In a future where most of the *in silico* experiments are outsourced, do we even need to train students in bioinformatics? Yes, and now more than ever.

Educating the next-generation of bioinformaticians

For the past 3 years, I have co-taught a large undergraduate genetics course, and the most popular part by far is the two-lecture section on bioinformatics. This makes sense, given that nearly all of the 1,100 students were reared in the age of Google and Facebook, and many appear to be more comfortable with digital devices and the online arena than they are with the biological world. So, when I present to the class a hypothetical future where DNA sequencing and genome

analyses are performed with smartphones, where physicians routinely pull up patients' genetic data like they were blood pressure results, their heads pop up from the social media vortex, and I have their undivided attention. And their attention I can keep provided I focus on the breakthrough changes that technological advancements will bring to the field of bioinformatics, and what that will mean for science, healthcare, and society as a whole. But woe betide me if I start mentioning the mathematical, genetic, and computational theories underlying common bioinformatics techniques.

> *"... I believe everyone would agree that an even more pressing issue is getting the proper bioinformatics tools and training to the scientists and students who need them."*

Therein lies the problem. At its surface, bioinformatics is a relevant, relatable, and stimulating discipline. At its heart, however, it is a dry, dense, and challenging topic to teach. If I describe the capabilities of user-friendly software like Geneious, the students are interested and engaged. "This is awesome, Professor Smith! I didn't realize that I could explore all of these cool genomes right from my laptop without leaving the couch and with only rudimentary computer skills." Bring up the finer points of the De Bruijn graph assembly method or Bayesian phylogenetics and half the class is heading for the door; and even some good old bioinformatics humor—"I hope everyone is having a BLASTX"—would not bring them back.

As you have already gathered from this essay, I neither have the desire nor the expertise to teach an in-depth technical course on bioinformatics, and to the best of my knowledge, neither does anyone else in my department. The only high-level bioinformatics courses offered at my institute are in computer science. Consequently, most of the biology students from my university have had limited exposure to bioinformatics when they graduate. This is both unfortunate and troubling given the now pivotal role of bioinformatics in the life sciences. And although some universities do offer more thorough bioinformatics training under the umbrella of biology, many others appear to be struggling to meet the growing educational needs of their students. In some ways, this reflects a broader trend across society of students and workers needing better programming skills.

The idea that coding is a crucial asset in today's workforce and one that can help students develop strong problem-solving capabilities has spurred various educational programs, including Apple's Everyone Can Code Initiative. On January 19, 2018, Apple announced that 70 colleges and universities across Europe have adopted their Everyone Can Code curriculum, which is a full-year, comprehensive course designed by engineers and educators to teach coding and app design to students of all levels. Other initiatives, such as Girls Who Code, are aimed at attracting more women and girls into computer-related disciplines, including bioinformatics. A recent study found that women are underrepresented in the field of computational biology [6], more so than in biology as a whole. Thus, one of the key priorities for the field of bioinformatics should be recruiting more female students, teachers, and scientists.

As more and more people start learning and using bioinformatics methods, scientists will be faced with the dilemma of who precisely qualifies as a bioinformatician. Some have taken a hard stance on this issue, arguing that biologists who merely use bioinformatics tools to perform analyses but do not contribute to the conception of such tools (biologists like myself) should not be considered bioinformaticians [7]. Instead, they posit that "bioinformaticians are scientists who develop and conduct research based on a bioinformatics approach ... who understand the underlying 'mechanics' of bioinformatics or, more realistically, an aspect of bioinformatics (genomics, protein structure predictions, phylogenetic models, etc.)" [7]. But what, then, do you call a scientist who spends most of his or her day employing, but not developing, bioinformatics tools?

In the past, I have rallied against such a narrow definition, suggesting that we need to broaden our definition of what it means to be a bioinformatician, not restricting it to only those who develop software or design and maintain data resources [8]. Although I am no longer as adamant on this front, I still believe the term bioinformatician should, in some cases, include scientists who use computers and bioinformatics programs to address fundamental questions in biology, even if those scientists are not expert programmers themselves. Let us not forget that Steve Jobs was a terrible programmer, but most would agree that he had an amazing knack for understanding technical concepts. Craig Venter is celebrated in the fields of synthetic biology and genome sequencing—fields where coding and biology blend together—yet, to the best of my knowledge, he is neither an expert programmer nor a software developer. But no matter how strictly or loosely one defines a bioinformatician, I believe everyone would agree that an even more pressing issue is getting the proper bioinformatics tools and training to the scientists and students who need them.

Concluding thoughts

Living in these high-tech times, it can be hard to gauge and appreciate just how fast and fundamentally the world is changing. Forgot your wallet? Tap your phone to pay. Sharing a private moment? Post it to hundreds of online followers. Lonely? Swipe to the right. Broke? Join Uber. Given the ways technology are transforming our everyday lives, it is to be expected that how we train for and carry out scientific research will also change, especially in disciplines that are technologically focused, like bioinformatics. As the influence of genomics on life and research grows ever greater and new genetic engineering techniques take hold, it is hard to see how bioinformatics could not become one of the leading scientific disciplines of the future. Although only in its infancy, the field has already helped to generate multiple subdisciplines, such as systems biology.

> *"But as it flourishes, bioinformatics will also become more susceptible to the very same problems and concerns affecting Silicon Valley today."*

But as it flourishes, bioinformatics will also become more susceptible to the very same problems and concerns affecting Silicon Valley today. Will time and initiative bring gender parity to the field of bioinformatics? Will the bioinformaticians of tomorrow be slugging it out in the gig economy, getting paid by the assembly? Or will automated computers and pipelines make their training

obsolete? Will the private companies that produce, house, and analyze genetic data use them ethically? Will these companies' hidden algorithms record our every alignment, BLAST, and assembly and use them for targeted marketing? The answers to these questions are to be determined, but surely the more the scientific masses have access to and control over their bioinformatics data and software, the better it will be for us all.

Acknowledgements

DRS is supported by a Discovery Grant from the Natural Sciences and Engineering Research Council (NSERC) of Canada.

Conflict of interest

The author declares that he has no conflict of interest.

References

1. Smith DR (2015) Buying in to bioinformatics: an introduction to commercial sequence analysis software. *Brief Bioinform* 16: 700
2. Vincent AT, Charette SJ (2014) Freedom in bioinformatics. *Front Genet* 5: 259
3. Rutherford K, Parkhill J, Crook J, Horsnell T, Rice P, Rajandream MA, Barrell B (2000) Artemis: sequence visualization and annotation. *Bioinformatics* 16: 944 – 945
4. Kumar S, Stecher G, Tamura K (2016) MEGA7: molecular evolutionary genetics analysis version 7.0 for bigger datasets. *Mol Biol Evol* 33: 1870 – 1874
5. Okonechnikov K, Golosova O, Fursov M, UGENE Team (2012) Unipro UGENE: a unified bioinformatics toolkit. *Bioinformatics* 28: 1166 – 1167
6. Bonham KS, Stefan MI (2017) Women are underrepresented in computational biology: an analysis of the scholarly literature in biology, computer science and computational biology. *PLoS Comput Biol* 13: e1005134
7. Vincent AT, Charette SJ (2015) Who qualifies to be a bioinformatician? *Front Genet* 6: 164
8. Smith DR (2015b) Broadening the definition of a bioinformatician. *Front Genet* 6: 258