During the last ten years, the number and diversity of formal programs in bioinformatics and computational biology have grown dramatically to meet the burgeoning demand for people skilled at wrangling and making sense of biomedical data. According to our count, since 2006, certificates have more than doubled; PhD programs and undergraduate majors have increased about 80 percent; and MS programs are up nearly 60 percent.

How universities are struggling to define core competencies, adapt to big data, and tailor curricula in this constantly changing interdisciplinary field.
To assess the current educational landscape, *Biomedical Computation Review* talked to directors of graduate and undergraduate programs in this space. The emphases of these programs range widely as do their titles, which include not only bioinformatics, biomedical informatics and computational biology but also health data science, biomedical data science, quantitative biomedical science, and various combinations of these titles with genomics. Despite the diverse names, all of these programs encompass bioinformatics in its broadest sense (the use of computation and statistics to gather, store, analyze, interpret, and integrate data to solve biological problems).

Two key themes emerged. First, there is more than ever to learn. Students have to grapple with the explosion of new technologies and data types. The big data revolution has also upped the ante on computational skills and heightened the emphasis on statistics. “You can’t expect anyone but a superman or superwoman to get all of that knowledge out of graduate school,” says **Michelle Dunn**, PhD, a senior advisor at the National Institutes of Health (NIH), who is involved with the Big Data to Knowledge (BD2K) training initiatives. Second, programs suffer from considerable heterogeneity—both in what and whom they teach. Lacking
formal guidelines, educators have largely pieced curricula together based on local needs and resources. Plus, educators face the daunting challenge of jointly teaching biologists and computer scientists—who come from vastly different cultures with highly variable skill sets.

Bioinformatics educators are confronting these issues on several fronts. Efforts are underway to define the core competencies of the discipline and to recommend key changes for the big data era. Educators are also sorting programs into distinct groups by trainees’ goals—and tailoring curricula accordingly. “There’s so much work to be done that we need people across the spectrum,” Dunn says. Educators are also experimenting with new ways to bridge the divide that separates researchers with disparate training backgrounds; and they are leveling the playing field by infusing more interdisciplinary backgrounds; and they are leveling the playing field by infusing more interdisciplinary training at the undergraduate level. Finally, several initiatives are creating a plethora of publicly available educational resources to meet training needs at all levels.

DEFINING CORE COMPETENCIES

Bioinformatics is a young, highly interdisciplinary, and rapidly changing field—so it’s been difficult for practitioners to agree on a set of standardized curriculum guidelines. An early attempt to delineate core competencies appeared in a 1998 Bioinformatics paper by Russ Altman, MD, PhD, professor of bioengineering, genetics, and medicine, and the director of the Biomedical Informatics Training Program at Stanford University. Altman laid out proficiencies in five domains: biology, computer science, statistics, core bioinformatics, and ethics.

But a decade and a half later, when Lonnie Welch, PhD, was searching for formal curriculum guidelines sanctioned by the International Society for Computational Biology (ISCB), he was surprised to learn that there weren’t any. “I come from the computer science community, where they have a lot of standards and guidelines. And I just assumed that there were such things,” says Welch, professor of electrical engineering and computer science at Ohio University.

Welch—who directs graduate and undergraduate certificate programs in bioinformatics—offered to lead an ISCB task force to remedy this gap. The team surveyed core facility directors and combed job listings and curricula from individual universities looking for cross-cutting themes. The committee found high variability in what programs are teaching and what students come out knowing—leading to mismatches between students’ skill sets and employers’ expectations. “One thing the core facility directors told us is that oftentimes the skills they most need are lacking in the students they hire,” Welch says. “That’s a wake-up call for us.”

The team published curriculum recommendations in PLoS Computational Biology in 2014, including core competencies in five categories similar to Altman’s: general, computational, biology, statistics and math, and core bioinformatics (see the table above for specifics).

From this list, our interviewees highlighted several areas in which training programs are falling short. Casey Greene, PhD—who has helped to shape Dartmouth’s PhD program in quantitative biomedical sciences—points to weaknesses in statistics, software engineering, and biology training. “Some training programs are failing to teach statistics that are relevant for big data,” says Greene, now an assistant professor of systems pharmacology and translational therapeutics at the University of Pennsylvania’s Perelman School of Medicine. Just learning about ANOVA (analysis of variance) and t-tests isn’t going to cut it anymore, he says. Within computation, students are well-trained in programming and algorithms but lack the engineering skills needed to build robust, reproducible, and usable tools, he says. A subset of students also need more exposure to “real biology”—the kind of wet-lab experiences that “give you an idea of how many things you can screw up.”

In this era of large-scale collaborative research, programs also need to better emphasize general skills—such as project management, creative problem solving, and communication, says Li-San Wang, PhD, associate professor of pathology and laboratory medicine at the University of Pennsylvania. These skills are learned by working on research projects, says Wang, who chairs the interdisciplinary PhD program in genomics and computational biology. “These are things you can’t even teach in classes.”

Altman stresses the importance of core

<table>
<thead>
<tr>
<th>Skill Category</th>
<th>Specific Skills</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>time management, project management, management of multiple projects, independence, curiosity, self-motivation, ability to synthesize information, ability to complete projects, leadership, critical thinking, dedication, ability to communicate scientific concepts, analytical reasoning, scientific creativity, collaborative ability</td>
</tr>
<tr>
<td>Computational</td>
<td>programming, software engineering, system administration, algorithm design and analysis, machine learning, data mining, database design and management, scripting languages, ability to use scientific and statistical analysis software packages, open source software repositories, distributed and high-performance computing, networking, web authoring tools, web-based user interface implementation technologies, version control tools</td>
</tr>
<tr>
<td>Biology</td>
<td>molecular biology, genomics, genetics, cell biology, biochemistry, evolutionary theory, regulatory genomics, systems biology, next generation sequencing, proteomics/mass spectrometry, specialized knowledge in one or more domains</td>
</tr>
<tr>
<td>Statistics and Mathematics</td>
<td>application of statistics in the contexts of molecular biology and genomics, mastery of relevant statistical and mathematical modeling methods (including experimental design, descriptive and inferential statistics, probability theory, differential equations and parameter estimation, graph theory, epidemiological data analysis, analysis of next generation sequencing data using R and Bioconductor)</td>
</tr>
<tr>
<td>Bioinformatics</td>
<td>analysis of biological data; working in a production environment managing scientific data; modeling and warehousing of biological data; using and building ontologies; retrieving and manipulating data from public repositories; ability to manage, interpret, and analyze large data sets; broad knowledge of bioinformatics analysis methodologies; familiarity with functional genetic and genomic data; expertise in common bioinformatics software packages, tools, and algorithms</td>
</tr>
</tbody>
</table>

In a 2015 paper about adapting bioinformatics curricula for big data, Greene and his colleagues proposed these courses on statistical and computational challenges of big data. Adapted from AC Greene, KA Giffin, CS Greene, JH Moore, Adapting bioinformatics curricula for big data, Brief Bioinform, first published online March 30, 2015 doi:10.1093/bib/bbv018, by permission of Oxford University Press.
institutions have hired our bioinformatics graduates," he says. Though the wishlist of skills seems long, just having a defined list can help focus a curriculum. By emphasizing fundamental skills—rather than specific tools, problems, or data types—the guidelines also help narrow the learning space. Stressing fundamentals also "suits people up for the long haul," Altman says. "The world is going to be changing a lot in the next 30 years. Whatever you're looking at on the horizon, it's just a good idea to go back to fundamentals."

ADAPTING TO BIG DATA
The onslaught of big data is also putting additional demands on bioinformatics curricula—in particular in the realm of statistics. Training programs have to keep abreast of these developments or they risk producing graduates who aren't prepared for the job market.

Once viewed as a pairing of biology and computation, bioinformatics is increasingly recognized as a three-way pursuit: biology, computation, and statistics. This shift is reflected in departmental changes at many universities. In 2015, both Dartmouth and Stanford merged divisions of biostatistics and bioinformatics into new departments of biomedical data science. In recent years, Harvard's biostatistics department has also adopted a bioinformatics focus—with a new MS program in computational biology and quantitative genetics in full swing and an MS program in health data science in the works. What does statistics add to the mix? Whereas computer scientists focus on finding patterns in the data, statisticians worry about sorting out real patterns from spurious ones. "Statistics provides unique expertise in making inference by accounting for spurious ones," says Xihong Lin, PhD, professor of biostatistics at Harvard. "This is especially important when one deals with massive data, as more data means more noise and a higher chance for more mistakes."

But statistics courses for bioinformatics analysis. Students need to know about high-performance computing and parallel computing, for example. "Let's face it, however," says John Quackenbush, PhD, professor of computational biology and bioinformatics in the department of biostatistics at Harvard, "the amount of data we're dealing with in biomedicine is nothing compared to what the folks in Silicon Valley are amassing at Google or eBay or Facebook." So, the computational challenges in biomedicine are not trivial, but they're not as pressing as the statistical challenges, he says.

The era of big data will also require entirely new ways of thinking about data, says Sean Eddy, PhD, professor of molecular and cellular biology and of applied mathematics at Harvard. "We have to learn to interact with these massive datasets in an experimental fashion," he says. For example, you can simulate data that come from the null hypothesis as a negative control—if the statistical tools you're applying find a positive signal, then you know the approach is faulty. When approaching data from this empirical view, biologists actually have an advantage. "We're trained to deal with big black boxes where we can't see all the moving parts. We know how to do experiments to ask questions out of complicated systems," Eddy says.

Most statistics courses aren't teaching this approach yet, but it's coming, he says.

TAILORING THE CURRICULA
Some heterogeneity in training is warranted. Curricula need to be tailored to the degree (BS, MS, PhD, or certificate) and the trainees' end goals. Welch's task force is making these different needs explicit. In particular, they have pointed out that bioinformatics practitioners fall into three distinct groups: Bioinformatics users are bench biologists or physicians who use bioinformatics tools in research or patient care; bioinformatics scientists develop algorithms and pipelines to answer specific biomedical questions; bioinformatics engineers support science by building robust software and computational infrastructure. Some graduate programs are aiming to train engineers, others scientists, and still others (often at the master's level) are more focused on training users.

"One of the reasons that there is so much friction and tension in bioinformatics education is that we've tried to put bioinformatics into a single box," Welch says. "Just calling out these three categories helps us to move the conversation forward." Each group needs varying levels of depth and breadth across the different competencies. You can view each skill as lying on a continuum—and the depth that you need in each depends on which of the three groups you fall into, Welch says. For example, users and scientists are going to be further along the continuum of life sciences knowledge than engineers, whereas engineers will have more depth in software engineering and system administration. "So what we're working on now is: How do you specify the points along the different axes where a person roughly should be if they want to be a certain type of bioinformatician at a certain level of career and degree."

BRIDGING THE DIVIDE
One of the most vexing issues in bioinformatics education is the heterogeneity of...Bioinformatics practitioners fall into three distinct groups: Bioinformatics users are bench biologists or physicians who use bioinformatics tools in research or patient care; bioinformatics scientists develop algorithms and pipelines to answer specific biomedical questions; bioinformatics engineers support science by building robust software and computational infrastructure.
the students. Cross-trained students do exist, but they’re in short supply—and they tend to get siphoned off by the most elite programs. Even when admission requires firm grounding in both computation and biology, “There’s almost nothing you can assume in common among the whole incoming student body, even at the PhD level,” says Russell Schwartz, PhD, professor of biological sciences and computational biology at Carnegie Mellon University and codirector of their PhD program in computational biology, which is offered jointly with the University of Pittsburgh.

For most programs, students tend to come in with strength in one area—either biology or computation—but not both. To train up the other side, many programs offer short courses or boot camps such as “Programming for Scientists” or “Crash Course in Biology for Engineers.” The idea is to break down the barriers, Eddy says. For a biologist, a key barrier is writing a Perl or Python script. “So you need to hold their hand, give them example scripts, and convince them that this is actually not as hard as they might think. It’s not computer engineering; it’s just like pipetting—just do it,” he says. Once students get past their initial fears, they can go learn more on their own. Students are increasingly able to acquire new skills by taking massive open online courses (MOOCs), which offer user-friendly introductions to computer programming, statistics, and biology. (See “Skills Upgrades” on page 7 of this issue.)

According to Greene, students with biology backgrounds are often portrayed as mathematically challenged and thus harder to bring up to speed. But, he says, “We have a lot of smart people that come from molecular biology and a lot of smart people that come from computer science.” In fact, he believes that mastering the biology is the tougher job because of the field’s nuances. “You can teach someone a superficial amount of biology quickly. But it’s hard to give them enough training so that they will have an intuitive grasp of why the data that they are analyzing look weird,” he says. For example, Greene recounts how a student analyzing data noticed that one day’s worth of numbers looked strange. It took a field trip to the lab to figure out the explanation: A new individual had begun washing the glassware on that day and had likely left residual soap. “It’s amazing what a trip to the lab can reveal that data analysis won’t.”

Bridging the culture gap can be harder than bridging the skill gap. Computer scientists and biologists have different ways of thinking, and they speak different languages. This is where joint advising and joint research ventures can help. At the University of Pennsylvania, doctoral students in genomics and computational biology often have dual advisors—one from biology and one from a quantitative discipline. “This works really well because after students finish their coursework, they still...
get exposure and advice from both sides,” Wang says. In his courses at Ohio University, Welch uses team projects to foster interaction. “When it works, you see the computer scientists helping the biologists to program and the biologists explaining the molecular biology to the computer scientists—it’s just so fun to watch.”

STARTING EARLIER

To reduce heterogeneity at the graduate level, students need better undergraduate training. “Just like someone going into a physics graduate program could be assumed to have a certain amount of undergraduate training in physics and not have to start from zero, I’d like to see us get to the same point with bioinformatics and computational biology,” Schwartz says. We need more undergraduate programs in bioinformatics/computational biology; and more training in statistics and computer programming for all science majors, he says.

The University of Nebraska at Omaha was one of the first institutions to offer a major in bioinformatics—starting in 2004. At first, they just pieced the degree together from pre-existing courses in chemistry, computer science, and biology, says Mark Pauley, PhD, senior research fellow at the College of Information Science & Technology and one of the developers of the major. However, over time the program has developed a series of discrete courses in bioinformatics. The program prides itself on being comprehensive, Pauley says. The major requires 24 credits in bioinformatics, 24 in computer science, 16 in biology, 17 in chemistry, and 11 in math.

Stanford has offered an undergraduate major in biomedical computation since 2003. “We used to have a lot of students trying to craft their own programs. So a bunch of faculty got together and designed an independent major,” Altman says. The major comprises four math courses (including statistics), four computer science courses, three chemistry courses, three biology courses, two engineering courses, a physics course, and a “technology and society” course.

It’s tricky to pack so much into a four-year degree and hard to find faculty to build these programs—particularly at small liberal-arts colleges. “So there still aren’t a whole lot of majors in existence, though it’s growing,” Pauley says. Other universities have compromised by creating certificate programs or minors. For example, Washington University in St. Louis offers a minor in bioinformatics that comprises three biology courses (including a lab), two computer science courses, a statistics course, and a bioinformatics algorithms course. “It’s kind of pasted together, and we don’t have as many discrete courses in the field as we would like, but it helps prepare students for bioinformatics graduate work,” says Sarah Elgin, PhD, professor of biology.

An even deeper problem is that undergraduate science majors in general aren’t receiving adequate statistics and programming training. “I don’t think it’s tenable for people to be entering a sciences graduate program or graduating from a sciences undergraduate program and not have a solid grounding in statistics or not know how to write a basic computer program,” Schwartz says.

In particular, undergraduate programs in biology have been slow to add quantitative requirements—despite repeated calls for curricular updates. “It’s the usual story that many people who go into biology do so because they love science but are scared of equations,” says Cath Brooksbank, PhD, head of the training program at the European Molecular Biology Laboratory–European Bioinformatics Institute (EMBL–EBI) in the United Kingdom. “Biology is not portrayed as a quantitative science. But it is a quantitative science, and it’s becoming increasingly quantitative.” At Washington University, Elgin says they have added a statistics requirement to the biology major, but programming is still not required.

TEACHING THE BIOLOGISTS

Though biology education has been slow to adapt, the field of biology is rapidly changing. Before long, every biologist will have to be a bioinformatics user. So, educators are trying to embed bioinformatics, statistics, and computer science into biology education at all levels.

Before long, every biologist will have to be a bioinformatics user. So, educators are trying to embed bioinformatics, statistics, and computer science into biology education at all levels.
what a gene looks like, they don’t understand gene structure very well,” she says. In formal assessments, GEP students improved their scores on a genomics/genetics quiz, and reported gains in understanding the nature of scientific investigation on par with students who spent a summer in a research lab, according to a 2014 paper in *CBE Life Science Education*, by Elgin, Rosenwald, and others. Even more importantly, Elgin says, the students gained awareness of the vast amounts of data available and the importance of computers in extracting new knowledge from that data. “Hopefully we are also waking up those biology students, inspiring them to sign up for more math and computer science courses,” she adds.

Another way to add bioinformatics content is to link an existing biology course with an existing computer course. Such “in-concert teaching” is described in a 2014 paper in *PLoS Computational Biology* by Anya Goodman, PhD, associate professor of chemistry and biochemistry, and Alexander Dekhtyar, PhD, professor of computer science at California Polytechnic State University, San Luis Obispo.

Students attend separate lectures but collaborate on joint labs and projects. The computer science students write the programs, and the biology students specify the programming requirements and test the software—so the two groups learn how to work in a cross-disciplinary team.

One of the major stumbling blocks to bringing bioinformatics into biology is the lack of biology faculty trained in this area. “I was a full professor before I had a personal computer,” Elgin notes. So, Rosenwald has created a project, GenomeSolver (http://genomesolver.org), aimed at training biology faculty to use basic bioinformatics tools. “If the faculty don’t know this, then the students don’t get the exposure to this important way of thinking about biology,” Rosenwald says.

**POOLING RESOURCES**

With so much to cover and so many audiences to serve, bioinformatics educators are getting together to pool resources.

In 2012, educators founded the Global Organization for Bioinformatics, Learning, Education, and Training (GOBLET, http://www.my goblet.org). The goal is to connect trainers across the globe so they can share expertise and training materials. The group is working to establish global curricula standards and accreditations; and they have created a training portal where educators can deposit and find high quality lectures, exercises, and datasets for teaching.

Other organizations are at work building repositories of publicly available biomedical data. Though originally meant to facilitate bioinformatics research, these organizations are also playing a significant role in education. For example, EMBL–EBI maintains a comprehensive range of freely available molecular databases and the tools to share, analyze, and query data. “When I first joined the EMBL–EBI, the vast majority of users were bioinformaticians,” Brooksbank says. “But our user base has grown and diversified hugely since then. Our training program needs to cater to this diversity.” So EMBL–EBI offers online training as well as workshops for graduate students, postdocs, faculty members, and industry professionals.

MOOCs are also bringing bioinformatics education to a wide audience. Pavel Pevzner, PhD, a professor of computer science at the University of San Diego, offers six short courses in bioinformatics on the MOOC platform Coursera. These problem-driven courses can be taken with or without a programming component, opening them up to biology and bioinformatics students alike.

In a 2014 paper in *PLoS Computational Biology*, David Searls, PhD, an independent consultant, argues that “a sufficient number and variety of free video courses have made their way to the web that it is possible to obtain a reasonably comprehensive bioinformatics education on one’s laptop.” He has assembled a catalog of relevant online courses organized into virtual departments, such as math, computer science, and biology, and proposed comprehensive curricula for different groups (such as bioinformatics users versus engineers).

The availability of so many training resources takes some pressure off formal university programs. Programs don’t have to teach every student everything. Rather, they need to give students a firm grounding in the fundamentals plus the tools for lifelong learning. “What we’re trying to do by the end of the graduate program is to have people who are pluripotent—who can go many directions from there,” says Dunn.