COLUMN Negative results can be interesting — but they still hurt **p.321**

@NATUREJOBS Follow us on Twitter for the latest news and features **go.nature.com/e492gf**

NATUREJOBS For the latest career listings and advice **www.naturejobs.com**

IMAGES.COM/CORBIS

COMPUTING

Out of the hood

Biologists frustrated with wet-lab work can find rewards in a move to computational research.

BY ROBERTA KWOK

During her master's programme in genetics from 2005 to 2008, Sarah Hird dreaded going into the lab. She was studying subspecies of red-tailed chipmunks and had become discouraged and frustrated by the uncertainties of molecular-biology experiments. She spent six weeks trying to amplify repetitive sequences in chipmunk DNA as part of an experiment to identify genetic differences between populations — but to no avail. Hird tried replacing reagents, switching to a different machine for running the polymerase chain reaction and decontaminating the sample-preparation area. Nothing worked. And then, for reasons that she never quite deciphered, the technique suddenly started working again.

By the end of her master's, Hird had come to dislike working in a wet lab, and she decided not to apply for PhD programmes.

About six months after finishing her master's degree, while working as a part-time technician at Louisiana State University in Baton Rouge, she discovered a better direction. The lab's principal investigator had suggested that she learn a computer-programming language so that she could help with a simulation project. Hird, who had never programmed before, taught herself the language using a book and online tutorials, and quickly became engrossed. "Once I started, it was like an addiction," she says. She enjoyed developing algorithms, and she found the software-debugging process less frustrating than troubleshooting wet-lab problems. The work felt more under her control.

Hird decided to stay in science, but to focus on dry-lab research — which generally entails writing software and developing models to analyse or explain data. After gaining admission to Louisiana State University's PhD programme in evolutionary biology, Hird developed software programs to manipulate sequence data for her dissertation. To boost her computer-programming skills, she took an undergraduate computer-science course and enrolled in a summer software-development programme. This year, she started a postdoc on computational analysis of bacterial sequences at the University of California, Davis.

Hird's success demonstrates that a wet-lab biologist can make a smooth transition to the dry lab, even without extensive formal computer-science education. Resources for learning computational biology and bioinformatics abound: there are training websites, online forums and workshops. And motivated scientists with a knack for quantitative thinking and problem solving can gain the skills necessary to make the change during a PhD programme or postdoc. "A lot of people get scared about even trying," says Casey Bergman, a computational biologist at the University of Manchester, UK. "It's a much easier transition than most people think."

Dry-lab researchers enjoy lower overhead costs than wet-lab scientists and have the flexibility to start new projects quickly. They can also more easily continue to run experiments when they become principal investigators, whereas scientists in wet labs tend to move entirely into managerial roles. Computational biologists face their own problems — some feel that they are taken less seriously than wet-lab scientists, for example — but those who set their own research directions and establish productive collaborations can thrive. For a researcher who wants to conduct experiments with quick turnaround times, and would like the ability to work on a wide array of topics, the flexibility to do research from home on a ▶

computer and the option to move into jobs in software development or data analysis, the dry lab can be a promising career option.

LEARNING TO CODE

A wet-lab scientist without a programming background can take courses in computational biology or software programming at university. But for those who want to try out the dry lab without committing a lot of time, there are several do-it-yourself options. The website Galaxy (galaxyproject.org), for example, offers a user-friendly way to run computational-biology analyses. Users can upload DNA- or RNA-sequence data and use a point-and-click interface to run software, such as a tool to find regions of DNA in which promoters overlap binding sites for transcription factors. A molecular biologist might analyse genetic data from their wet-lab experiments or the sample data sets available on the website.

The site, developed by researchers at Pennsylvania State University in University Park, and Emory University in Atlanta, Georgia, also provides an interactive tour that demonstrates how to carry out analyses. Galaxy is good for beginners because it allows them to try bioinformatics tools without writing software programs, says Bergman.

To start programming, Hird suggests reading a book for beginners and doing the exercises. Websites such as Codecademy (codecademy.com) offer online tutorials. Aspirants might start with an easy language such as Python instead of more difficult languages such as Java or C++, says Pavel Tomancak, a developmental biologist at the Max Planck Institute of Molecular Cell Biology and Genetics in Dresden, Germany.

He notes that newcomers may stay more motivated if they can apply computational skills to real scientific problems rather than to the 'toy' exercises in a computer-science class. For example, a researcher who works with many image files could write a program to automatically perform processing steps, such as contrast enhancement, on thousands of images.

Biologists can practise their programming skills on websites such as Rosalind (rosalind.info), developed by researchers at the University of California, San Diego, and at St Petersburg Academic University in Russia. The site presents progressively harder bioinformatics problems and asks the user to write programs to solve them. An early problem involves converting a DNA string into its transcribed RNA strand; a more advanced exercise requires the user to find ways in which RNA molecules can bind to each other.

Users download sample input files that contain the data to be processed, run their programs on those data and copy the output to the site to confirm whether their answers were correct. Once users have completed an exercise successfully, they can view solutions that other users have written. Those who can make a bigger time commitment can pursue more-formal training options (see 'Classes in computing').

NOAH M. REID

"Once I started, it was like an addiction."

Sarah Hird

Beginners can also ask questions on online bioinformatics, sequencing and programming forums such as Biostar, SEQanswers and Stack Overflow. "If you're feeling a bit stuck, there's no shortage of help," says Neil Saunders, a statistical bioinformatician at the Commonwealth Scientific and Industrial Research Organisation in Sydney, Australia. To avoid receiving snide responses, newcomers should make their questions specific and describe their efforts to solve problems instead of simply asking others to write their code.

TIME FOR A CHANGE

Scientists seeking a transition to the dry lab can build their computational skills during their wet-lab PhD programme. They might consider using computational tools in their dissertation work, as well as seeking out a supportive mentor or committee member, says Hird. To gain an adviser's support, a doctoral student could explain how these skills would enable better research: for example, a graduate student in developmental biology could use programming skills to perform complex three-dimensional data visualizations. Researchers could also collaborate with dry labs to analyse their data, suggests Karen Cranston, a computational evolutionary biologist at the National Evolutionary Synthesis Center in Durham, North Carolina.

PhD graduates can land dry-lab postdocs without extensive programming experience. Bergman finds that applicants with the perfect blend of skills in biology and computer science are rare, so he is willing to hire and train biologists who do not know programming as long as they have strong quantitative skills. Nevertheless, applicants can improve their chances by using Rosalind, learning a statistical software package such as R or completing a computational project through a programme such as Google Summer of Code.

Computational biologists tend to enjoy more flexibility than their wet-lab counterparts. Researchers can often start a new project by downloading data and writing a program, instead of having to order reagents or spend months optimizing protocols. Costs are generally lower because dry labs do not require reagents or staff to maintain cell lines, and

FORMAL TRAINING

Classes in computing

Researchers seeking formal training in computational biology or bioinformatics can enrol in classes, workshops and mentoring programmes. Sarah Hird, a postdoctoral researcher in evolutionary biology at the University of California, Davis, gained computational skills during her PhD programme by enrolling in 2011 in Google Summer of Code. The programme, run by Google in Mountain View, California, funds project-proposal-based software-development projects by undergraduates and graduate students, and links participants with mentoring organizations. Hird learned the programming language Python and developed software to display features of DNA-sequence data in a tabular format, receiving guidance from two experienced programmers through e-mails, Skype calls and comments on her blog posts about the project.

Last month, researchers at the University of California, San Diego, began teaching a free online course on bioinformatics algorithms through the education website Coursera (www.coursera.org). They hope to repeat the course next year. Many institutions run computational-biology workshops, including Cold Spring Harbor Laboratory in New York, the National Evolutionary Synthesis Center (NESCent) Academy in Durham, North Carolina, and the European Molecular Biology Laboratory in Heidelberg, Germany. Karen Cranston, a computational evolutionary biologist at NESCent, suggests signing up for mailing lists to find out about upcoming workshops.

Software Carpentry, an international volunteer organization, offers software-development workshops for scientists worldwide. The two-day boot-camps help researchers learn efficient programming, such as managing software versions and setting up automated bug testing. Researchers wanting their institution to host a workshop should contact Software Carpentry, says Titus Brown, an advisory-board member and bioinformatician at Michigan State University in East Lansing. The organization can provide a 'pitch' presentation that contains information about benefits and costs, which the researchers can show to administrators to convince them to fund instructors' travel expenses. R.K.

results come back quickly, making it less risky to take research in a new direction. When a computer is the only equipment needed, one does not always have to work in one's office. Bergman often writes software or runs programs at home in the evenings. "That's what sustains my interest in the science," he says.

Dry-lab researchers can apply their skills to many problems. "It really expands what you can do," says Saunders. "If your speciality is analysing data, then the nature of the data is sort of secondary." Saunders has analysed the genes of Antarctic microbes, modelled protein structures to study how enzymes find the right substrates and searched for biomarkers to improve early detection of colorectal cancer. And dry-lab scientists have plenty of options outside academia: career opportunities are opening up at institutions such as museums, which need staff with computational skills to help to organize and share their data, such as biodiversity information, and at DNA-sequencing facilities. Bergman and Saunders know of dry-lab researchers who have gone on to work at genomics and online education start-ups in California's San Francisco Bay Area, or to develop mobile-phone apps or land data-analysis jobs at consulting companies.

Dry-lab scientists must ensure, however, that wet-lab collaborators do not view them only as technicians — they must make it clear that they are "actually a research scientist", says Saunders. "You're not just the computer guy with the magical program." Even when one is the principal investigator of a lab, it is easy to be perceived as just "providing a service", he says. So it is important that, when starting a collaboration, computational researchers discuss whether they will help to set the project's scientific direction. Saunders adds that it is fine to participate in some projects that are steered by wet-lab scientists, but dry-lab scientists should develop their own projects as well, such as developing computational tools.

Researchers who make the transition to the dry lab need not worry that their wet-lab experience is wasted. Dry-lab scientists still need to think like biologists and consider the complexities of the system being studied. Programmers tend to oversimplify complex problems, but biologists know that there are often exceptions to the rules, says Cranston. And some researchers combine wet-lab and dry-lab work rather than giving up the former entirely. "I don't really like to look at it like a switch. I look at it more like adding more tools to your toolkit," says Bergman. "It's not an either/or." ■

***Roberta Kwok** is a freelance science writer in Seattle, Washington.*

COLUMN

A faulty hypothesis

Carolyn Beans says that some of the most interesting results are negative ones — but it still hurts to be wrong.

MELANIE TAYLOR/SHUTTERSTOCK

I hold a strip of tinfoil above a bowl of water and ask the four children seated around me to make a hypothesis. "What do you think? Will it sink or will it float?" They have all sorts of ideas. "Float! It's too light to sink." "Sink! It's made of metal!" "Float! It's flat!"

I ask the students to circle their guesses next to the picture of the tinfoil in the hypothesis column of their worksheets. Then I drop the foil. The two students who circled 'float' begin flailing their arms in a wild dance of joy. The two who circled 'sink' erase their choice.

"Don't erase!" I beg them. "Just circle 'float' in the results column. Some of the most interesting scientific discoveries come when a hypothesis is wrong." They do not buy it. As we test more objects, the children continue to modify their hypotheses to match the results. One refuses to make a hypothesis until the results are known. Apparently, children do not like to be wrong, even in the name of science.

I thought of this elementary-school visit as I sat in my office last autumn, looking over the results of my summer research. I had tested whether three different plant species could reduce seed production in a jewelweed native to the United States, *Impatiens capensis*, by competing for pollinators.

One of the contenders was an introduced jewelweed from India, *Impatiens glandulifera*, with bright-magenta, nectar-rich flowers. I thought that surely this species would win over the local pollinators and reduce seed production in the native plant.

But it turns out that none of the competitors, either individually or as a group, had any influence on seed production in the native jewelweed. When I first saw these results, I thought how nice it would be to erase the past four months of work and start again.

As a graduate student, I find it hard not to feel like a failure when a hypothesis is incorrect. It was especially disheartening when I had committed an entire summer to an experiment, and there was no hope of restarting until the next growing season. Throughout the autumn and winter, I pondered. Was my sample size too small? Were my plots too close together? Should I have run the experiment for longer?

I came to the conclusion that my hypothesis was just wrong. Although this realization did not leave me dancing for joy like the children who circled 'float' (and I certainly would have danced had my hypothesis been correct), it did allow me to start identifying the positive aspects of my results.

First, I feel grateful that the experiment happened at all. Fieldwork is unpredictable and sample sizes are often demolished when study sites succumb to disease, unusual weather patterns or a stray lawnmower. Getting 700 potted plants of 4 species to flower all at the same time was no trivial task. For my success I thank good field assistants, decent weather and an influx of ladybirds that appeared just when it seemed that aphids would eat every last plant.

Second, I appreciate that my experiment was designed well enough to disprove my hypothesis definitively. I did not get what I expected, but I discovered something real. Now I can develop new experiments instead of wasting time redoing the same one.

Finally, I feel hopeful that the negative results mean that the story will turn out to be more nuanced and interesting than I initially expected. For example, this and other experiments suggest that even without influencing seed production, competitors may alter natural selection on the native jewelweed by causing pollinators to favour different-sized flowers.

As for the kids, they still want to be right. But what they want most is to do another experiment, and then another. They never tire of dropping objects into the water and thinking up reasons that one will float and another will not. They may not own up to their incorrect hypotheses. But they are learning that there is more to science than confirming one's expectations. ■

***Carolyn Beans** is a biology graduate student at the University of Virginia in Charlottesville.*