My brilliant career

Andrew Conway

Where can a mathematics education lead you? In this series, mathematics and statistics graduates from Australian universities write about their careers, proving that the world is their oyster. The endeavours of Andrew Conway, founder and former director of Silicon Genetics, eventually let him to cross to the "Dilbert dark side."

A mathematician starts a company

When I did my PhD in Mathematics, Dilbert was just becoming popular. I was a great fan, and strongly believed in the basic concept that the people who "do the work" — scientists and engineers — are the only people who were really useful for society. My friends considered me as hooked in academia — a PhD in Mathematics (working in combinatorics), and a second, recidivist, PhD in Electrical Engineering (working on helicopter navigation). Even when I carefully considered my strengths (quantitative science, computers) and what fields they could be used in and decided upon biochemistry, I returned to Stanford University to do it. So how did I move to the Dilbert dark side of running a company and all that implies?

There were of course many factors, but the strongest was seeing how the science of biochemistry was being advanced by major engineering efforts. As an example, in 1995 when I started in biochemistry, people involved with sequencing the human genome were looking at the output of machines and — by eye — stating what they thought the relevant base would be. The problem is, this needs to be done several billion times. Reliably. Accurately. Quickly. Preferably with quantified absolute confidence estimates. And, of course, cheaply — while there was a lot of money available for the human genome project from governments around the world, it dwindled to a matter of cents when divided by several billion. This was just one of the many engineering challenges involved in getting the sequences for the human and other genomes.

For those who are not familiar with the human genome project, it was a multinational project to basically measure the complete DNA sequence (think chemical formula — for a molecule with billions of atoms) of a human. Once obtained, it was a useful resource for life scientists. The human genome project was the most well known of such life science engineering projects, but there were actually many others.

After working in this field for a while, I realized that as an engineer I could do more for science than I ever could as a scientist. I also realized that for many types of tools, it was more useful to make them in a company than in academia. One idea that particularly interested me was making software to analyse the results of a particular type of experiment called "Gene Expression". Some colleagues had done a wonderful experiment, the first of its kind, where they measured the activity (by some definition) of almost every gene in some yeast cells while they grew and reproduced. This generated a table of numbers with six thousand odd rows — each representing a gene — and sixteen columns — each representing a time point. They came to me — as the local mathematician — and explained what they had done, and asked if I had any ideas what

to do with the wonderful but daunting table of a hundred thousand numbers they had generated. I looked around for data mining tools to help, couldn't find any sufficiently specialised, grumbled a bit about how someone should do something about it, and then realised that someone should be me.

A few years later the company I founded, Silicon Genetics had fifty odd people and a successful product in an expanding market. While my heart was still in development, I had to spend most of my time on other things — the Dilbert dark side. I even sometimes recognised myself in the butt of various of the strips.

If you build a better mousetrap, the world will not flock to your door. Most of the world won't know where your door is, won't realise that you even have a mousetrap, and don't really understand or care that yours is better. Some uncooperative people won't even have a problem with mice!

Fortunately the way to sell software to scientists is to get other scientists to show them how to use the software to solve their problems rather than showing pictures of scantily clad smiling teenagers on television waving the product around. But while our sales and marketing staff were scientists, we still had very similar issues to most businesses. We had to tell people we existed, explain why our product could solve their problem (and why better than any other product), teach people how to use it, support it, collect money, pay salaries, taxes, health insurance, etc., and generally keep a group of fifty odd people working together smoothly. I found new respect for good salespeople, marketeers, accountants, lawyers, and administrators, all of whom could make a big difference in how well a company worked, and how happy the customers and employees could be. Of course, I did also meet many incompetent ones after whom the Dilbert characters were modelled, but there are many competent, hard working, dedicated ones as well.

I also gained a great respect for middle management. The upper management gets the glory (and the money), while the middle management has to deal with lots of little annoying problems. Also, your manager is always dumber than you. Or certainly appears that way in many contexts. Suppose you have been working on something — in your field of speciality — for the past week, and you need some decision. You go to your manager, spend five minutes explaining to this less-of-a-specialist-in-your-area-than-you what you have been thinking about for a week, and expect a quick sensible decision. Sometimes the manager will have experience, extra information, or even just the authority to make that decision. Often the manager will not completely understand the problem, but still have to make a decision or suggestion.

Working out what a software product should actually do is surprisingly difficult, and many experienced software developers will tell you it is the main task (much of this is, of course, user interface). Of course there are times when working out how to do it is also important. Mathematics turned out to be very important in this.

Data mining software like ours contains of course a lot of statistics. Clearly one needs to understand the statistics being used, but it is actually important to understand the statistics that is not being used as well — otherwise, how could you work out what new approaches to use, or how to solve some new problem? Even when you have worked out what statistic you want to calculate, the logistics of how to calculate it often required a lot of mathematics to do efficiently or without rounding problems. I only spent a tiny portion of my time really using non-trivial mathematics — but if I couldn't do it, then I would have spent the rest of my time doing the wrong thing!

I did find that I had to learn a lot more statistics. As a mathematician I of course knew a fair bit of statistics, but not nearly enough. I'm not a statistician, but I came to

the understanding that any sentence starting with "I'm not a statistician, but ..." is almost certainly subtle nonsense. Statistics is often considered unfashionable or boring as an undergraduate course — but when we had user group meetings, the statistics talks were always the most popular.

Eventually, Silicon Genetics became a mature, successful business, and Agilent Technologies purchased it. Now I am back to working on various scientific problems — none directly mathematics, but all using it.

To me, mathematics is a language. A skilled mathematician is articulate in a language that forms the basis for much of the work that I do. It is important to be fluent in order to think clearly and communicate in these fields. I rarely think of myself as "using" mathematics — rather I just think and talk in mathematics.

E-mail: andrewFromAMSGazetteReader@greatcactus.org