First off I should say that my career path was not planned or plannable, so you can’t follow in my footsteps except to go where there is no path.

From a young age (7 years old), my vocation was to be a scientist, and I quickly settled on theoretical physics because physics is everything - any explanation of the natural world eventually bottoms out in the laws of physics (yeah, I’m a hard-core reductionist and proud of it). At high school and college, I was pretty good at math and physics, but once I started my PhD I got a shock because I discovered that many people were much smarter than me. A few months after starting my postdoc I concluded that I would always be a mediocre physicist and would never make much of a contribution, so I quit. That was very difficult because physics had been my life’s dream, but in retrospect physics was the wrong field for me; I’m more of a natural programmer than a natural mathematician. Like many failed physicists in the mid-80s, I got into the software business and ended up starting my own company in San Francisco. I sold the business to Intel in 1999, and in 2001 I was burned out and quit with no idea what to do next.

I didn’t want to start another business, which is what most entrepreneurs do, and I didn’t want to retire and play golf (or more likely tennis). I knew about the human genome project and the importance of software algorithms - it was very appealing to me as a software guy that you could do important things just with strings of letters without knowing all that tedious biochemistry. Biology=strcmp(). So while it didn’t occur to me that I might actively work on that stuff, I thought it would be fun to learn something about it, and crashed a seminar at UC Berkeley and met a newly-hired professor named Kimmen Sjolander. She was looking for students to help her code up some algorithms for summer research experience, and I volunteered. The result was a multiple alignment program called SATCHMO (Edgar and Sjolander 2003), which worked pretty well but was no better than CLUSTALW according to BALIBASE, which was the only available benchmark at that time.

Being a competitive guy, I was dissatisfied with that, and started a systematic project to figure out what did and didn’t work in multiple alignment algorithms, because it seemed to me that the programs had many ideas in them, but it wasn’t clear which of the ideas helped or hurt the results. At that time, Gotoh’s PRRx had the best accuracy (except maybe for T-Coffee; I don’t remember) but the code was unfriendly and slow and hardly anybody used it. So I started from the PRRx algorithm and systematically varied the elements of the algorithm with all the alternatives I could find in the literature or dream up myself. That was the line of work that led to MUSCLE (Edgar 2004). (As an aside, the MAFFT people followed a similar strategy independently, and it’s remarkable how much the original MAFFT and MUSCLE v1 resembled each other. MAFFT got published first and was better than my first
attempt; I had to ‘borrow’ a couple of their ideas in order to do better than MAFFT (Katoh et al. 2002). For example, we both had k-mer counting to build the first tree, but I was using 3-mers in the usual alphabet and they used 6-mers in a compressed alphabet, which gave slightly better results.

So MUSCLE was a result of some financial independence I had after selling my business. I’ve supported myself and my research from my savings for the past decade. You can think of me as unemployed, independent and/or a gentleman scholar of modest means (like, say, my fellow-countryman Charles Darwin). It’s hard to know how people have perceived my unconventional status. Everyone feels misunderstood and dismissed by peers sometimes (reviewers, editors, conference organizers), so it’s impossible to say whether I would have been better accepted if I had a conventional affiliation. MUSCLE has been helpful because many people have heard of it. That gave me some street cred early on, and surely helped open other doors.

I don’t see any lessons here to help young scientists, so in an attempt to avoid becoming a bad influence I will defer to a more accomplished scientist, Richard Hamming, who has much better advice in his lecture, “You and Your Research” (Hamming 1986)

REFERENCES


