Learning to think like a bioinformatician: a reflection on 2026

I came into this Honours year with a biochemistry and microbiology background and frankly no coding experience at all. I hadn't written a line of Python or touched the command line before starting at the Division of Human Genetics. Looking back on it now, that gap has probably shaped my learning more than anything else.

The steepest part of the curve wasn't any single module it was the constant translation problem. I understood biology sure, but I had to learn to think in a completely different sort of language to reach it pipelines, scripts, statistical models, and a command line that really didn't forgive typos at all. Early on every assignment sort of felt like two assignments, one in genetics, and one in "how do I even get the computer to do this."

What changed things was working on real, messy data instead of textbook examples. I mean, stratifying TCGA-GBM patients by sex and survival time, then running DESeq2, GO enrichment, and KEGG pathway analysis, was kind of the first time the pipeline stopped being an abstraction and started being a tool I understood well enough to make choices with— like, which comparisons felt right, which outcomes were worth chasing, and which ones weren't. Then the crystallography work on Rv0480c pushed a different kind of precision: getting a covalent modification modelled correctly isn't something you can approximate your way through, and that same discipline kind of leaked over into how I approached later analyses as well.

If I'm honest, the biggest shift wasn't technical, it was like learning to sit with not understanding something, yet without acting like that was some proof I shouldn't be here. Bioinformatics kind of rewards the folks who can tolerate being disoriented for a while, not just the ones who instantly get it. Over the year I've gotten measurably better at that, though it didn't feel smooth most of the time.

What I'd tell the version of me that started in this program: the coding catches up faster than it looks like it will. The biological intuition you already have is the harder part, and honestly it's the one that makes the technical side worth learning in the first place.