Why Coding?

The Pedagogical Case for Computational Literacy in Biochemistry and Molecular Biology

PyBMB
Python for Biochemistry and Molecular Biology

Wally Novak
CodeBMB Workshop

By the End of This Presentation, You Will Be Able To...

- Define **computational literacy** and describe its three interconnected components — material, cognitive, and social
- Articulate **why computational skills are essential** for the next generation of biochemistry and molecular biology students
- Identify **three major barriers** that prevent faculty from incorporating coding into their BMB courses
- Explain how the **TPACK framework** maps the knowledge faculty need to teach coding effectively in a BMB context
- Describe at least **two pedagogical strategies** — including notebook-based learning and live coding — that make teaching coding accessible to beginners

The Elephant in the Room

Should We Be Teaching Coding to Life Scientists?

- Many of us were trained without coding
- Many of our students will work in fields that require it
- Most BMB faculty believe coding is important — but don't teach it

95% of faculty survey respondents believe coding is important...

62% have NEVER run a coding activity in their courses

Yet faculty report insufficient preparation to teach it

Training faculty to teach computation is essential for sustaining the research pipeline

What Is Computational Literacy?

It's More Than Just Writing Code

According to diSessa…

a new literacy emerges when there is an **adoption** of some tool or way of reasoning by a **broad cultural group**

Material	Cognitive	Social
The ability to read and write code	Applying computation to analyze and solve problems	Communicating through computation — e.g., commenting code, collaborating, visualizing

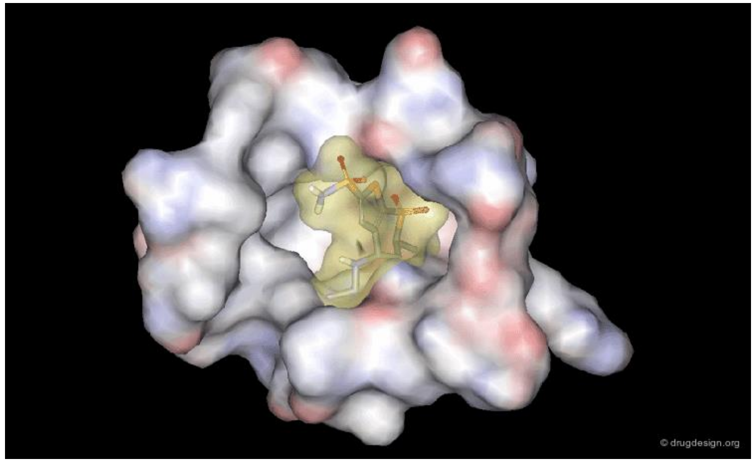
Training in all three aspects is widely regarded as a 21st-century skill and improves student learning gains.

(diSessa, 2001; Weintrop et al., 2016)

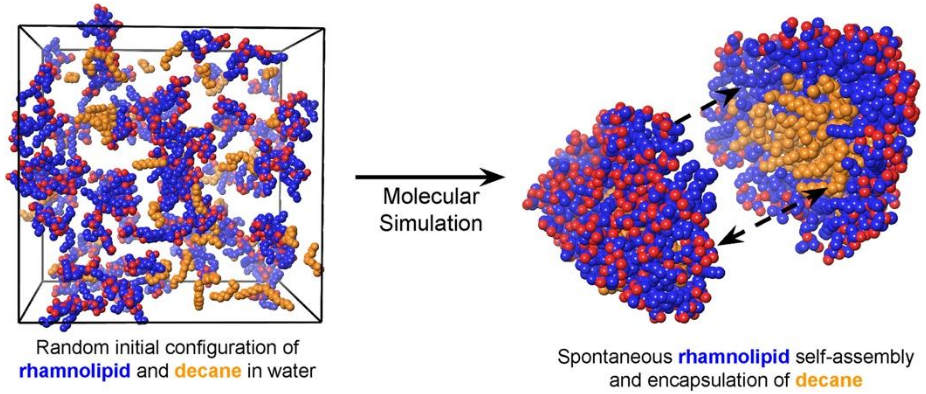
Why Now?

Computational Literacy Is a 21st-Century Requirement

- Bioinformatics, computational biology, protein design, and NGS analysis all rely on computational skills
- ASBMB recognizes computational skills as necessary for BMB curricula
- Computational literacy allows scientists to develop models and test hypotheses rapidly and inexpensively
- AlphaFold, AI-driven drug discovery, and large-scale genomics are already the present
- Make America AI Ready (DOL) – AI literacy for the workforce



https://www.drugdesign.org/chapters/molecular-docking/



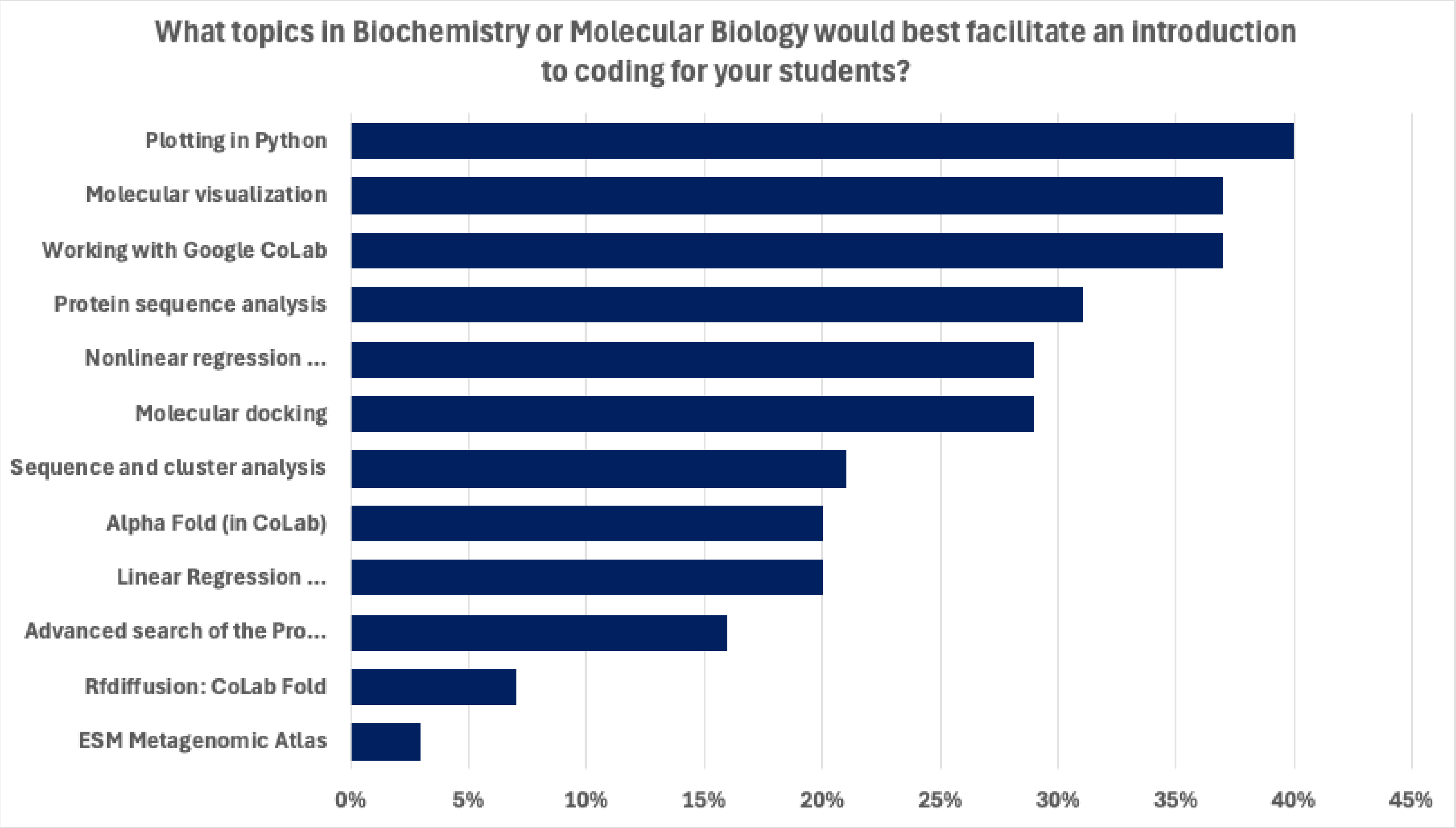
https://doi.org/10.1016/j.cocis.2023.101760

Big Data Moments in BMB

- 1990s: BLAST & GenBank
- 2000s: Human Genome
- 2010s: NGS Revolution
- 2020: AlphaFold
- Now: AI in drug design

What Our Community Says It Needs

Survey of BMB Faculty (January 2025)



Why Hasn't Coding Made It Into Most BMB Classrooms?

Barriers to coding in the BMB classroom and how we are addressing the issue

- Coding is perceived as complex
- Access to coding is not inclusive
- We are reluctant to teach what we don’t know

https://goeshow.com/the-event-managers-guide-to-diversity-equity-and-inclusivity/

Carbone et al. (faculty barriers), Van Wyngaarden et al., Derus & Ali (computing facilities)

Barrier 1: Coding Feels Hard

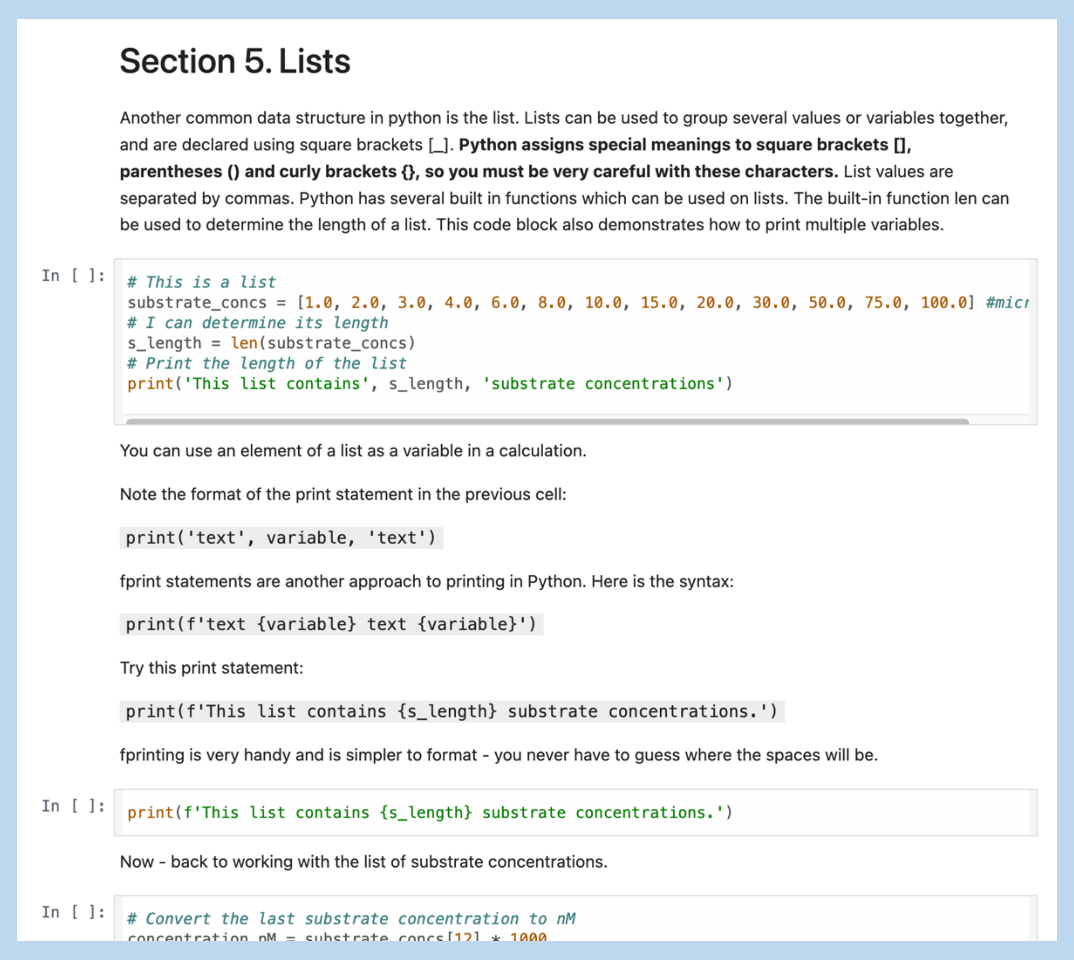
"I don't even know where to start"

The challenge:

- Students must learn syntax AND how to apply it to specific tasks
- Perception of difficulty and complexity remains high

The solution — notebook-based learning:

- Colab allows code to be provided in ***ready-to-run*** format
- Code is run in small sections → incremental progress
- Students learn coding while performing contextual BMB tasks
- Notebooks are simple to run and modifiable for new problems



Section 5. Lists

Another common data structure in python is the list. Lists can be used to group several values or variables together, and are declared using square brackets [_]. **Python assigns special meanings to square brackets [], parentheses () and curly brackets {}, so you must be very careful with these characters.** List values are separated by commas. Python has several built in functions which can be used on lists. The built-in function len can be used to determine the length of a list. This code block also demonstrates how to print multiple variables.

In [ ]:
```
# This is a list
substrate_concs = [1.0, 2.0, 3.0, 4.0, 6.0, 8.0, 10.0, 15.0, 20.0, 30.0, 50.0, 75.0, 100.0] #micr
# I can determine its length
s_length = len(substrate_concs)
# Print the length of the list
print('This list contains', s_length, 'substrate concentrations')
```

You can use an element of a list as a variable in a calculation.

Note the format of the print statement in the previous cell:

`print('text', variable, 'text')`

fprint statements are another approach to printing in Python. Here is the syntax:

`print(f'text {variable} text {variable}')`

Try this print statement:

`print(f'This list contains {s_length} substrate concentrations.')`

fprinting is very handy and is simpler to format - you never have to guess where the spaces will be.

In [ ]:
```
print(f'This list contains {s_length} substrate concentrations.')
```

Now - back to working with the list of substrate concentrations.

In [ ]:
```
# Convert the last substrate concentration to nM
concentration_nM = substrate_concs[12] * 1000
```

McGettrick, The Computer Journal, Laakso, et al. Journal of Information Technology Education. Innovations in Practice, Ochkov, et al. VI International Conference on Information Technologies in Engineering Education

Barrier 2: Access and Equity

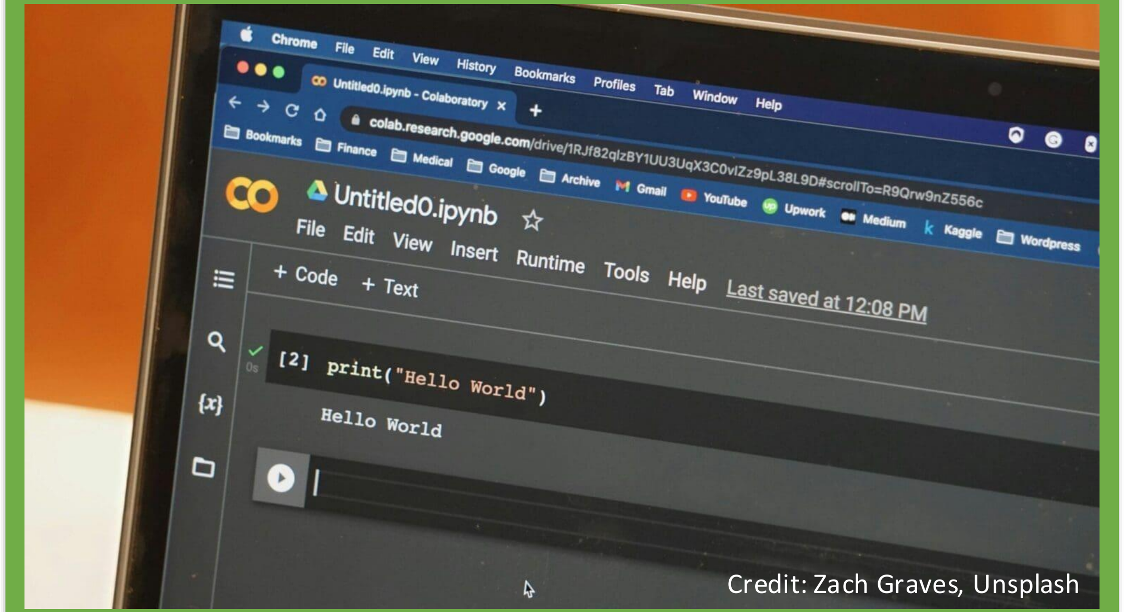
"Not everyone has the same resources"

The challenge:

- Computer access at higher education institutions varies widely
- Some students cannot afford their own device
- School labs may use Mac or Windows, disenfranchising some users
- Poorly functioning facilities detract from learning to code
  (Derus & Ali, ICCIE 2012)

The solution — cloud-based coding:

- Google Colab runs in any browser, so no installation needed
- Works on any operating system and most devices
- Free! No cost barrier for students or institutions



Credit: Zach Graves, Unsplash

Students at any institution can solve interesting biological problems for nearly no cost!

Barrier 3: Faculty Confidence

"I can't teach what I don't know"

The challenge:

- Lack of instructor confidence
- Lack of time, training, and professional development opportunities
- Most BMB faculty have little experience training others to code

The solution — turnkey materials + community:

- Pre-built Colab notebooks are simple to run and modifiable
- Live coding: instructor writes and runs code step by step
- A growing community of BMB faculty facing the same challenges
- You don't need to be an expert, you just need to be one step ahead

Credit: Wonderlane, Unsplash

Carbone et al. *Higher Education Research & Development,* Van Wyngaarden et al. *Trends in Higher Education*

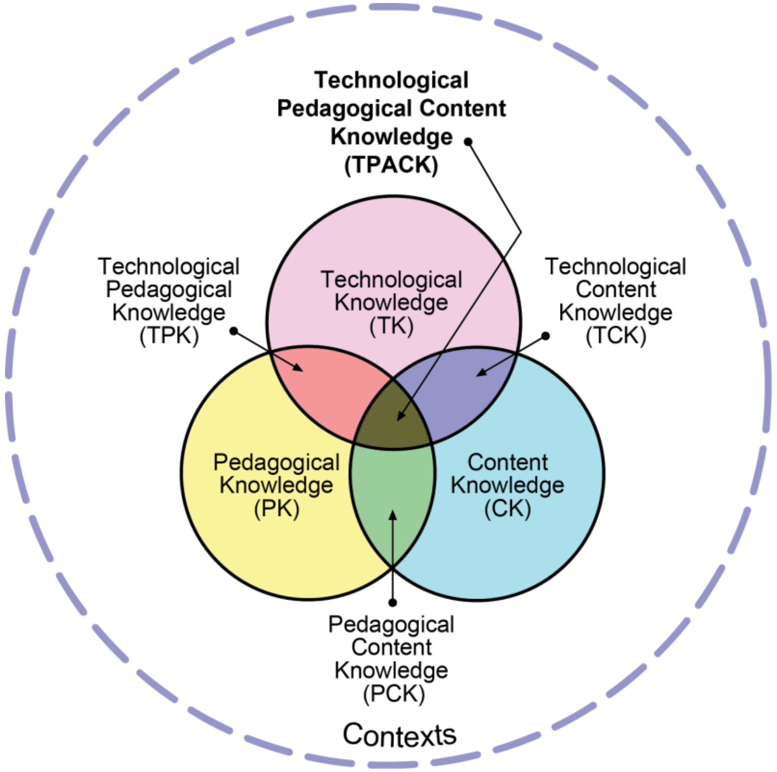
What Do You Actually Need to Know?

The TPACK Framework — Mishra & Koehler (2006)

PK - Expertise in teaching:
active learning, flipped classrooms, live coding, scaffolding

CK - Expert BMB knowledge:
enzymes, proteins, genomics, biochemical pathways

TK - Knowledge of technology:
Colab, GitHub, Python, BLAST, visualization tools



Important intersections:
TCK = using Colab + Python for BMB tasks | TPK = knowing HOW to use these tools to deepen student learning | TPACK = all three together

Where This Workshop Fits in TPACK

Today you will learn...	TPACK Area	How
GitHub + Colab + Python	TK	Introductory Lectures and Demonstrations Notebook 001
Python for BMB data analysis	TCK	Notebooks 002-003
Linear regression workflow	TCK, TPK	Notebook 004
Developing new notebooks for your class	TK, TPK, TCK, TPACK	Future Workshops
Using AI tools responsibly for generating assignments and engaging students	TK, TPK, TCK, TPACK	Future Workshops

Our goal: provide faculty with the TPACK of BMB coding to increase student learning.

How This Workshop Is Designed

Pedagogy in practice — four guiding principles

1 Context First

Concepts taught using BMB examples: Michaelis-Menten kinetics, Bradford assays, protein sequences.

2 Incremental Complexity

Introduce online resources.
Run code in small sections.
Scaffolded notebooks allow modification without starting from scratch.
Every participant makes progress.

3 Live Coding

Instructor writes and runs code step by step.
Makes the process of coding visible.
Demonstrated to make coding easier for novices to understand.

4 You Leave With Something Real

A series of notebooks that can be adapted to your own research or teaching.
More development in future workshops.

What You Will Leave With

- ✅ A working Google Colab + GitHub setup
- ✅ Foundational Python skills grounded in BMB examples
- ✅ Complete notebooks you can use in own research or teaching
- ✅ Membership in a growing community of BMB coders

You don't need to become an expert coder.
You just need to become confident enough to take the first step with your students.