CrossTalk Bootcamp

Benjamin Sanchez-Lengeling

Session 1, Intro to ML for molecules
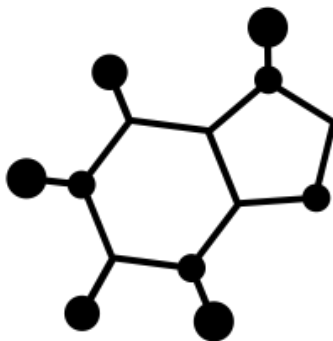
VECTOR INSTITUTE

UNIVERSITY OF TORONTO

Panorama

Coding in '2025

Exploratory
Data Analysis
With Molecules

DEL data

Panorama

Coding in '2025

Exploratory
Data Analysis
With Molecules

BB2
BB1
BB3
Tag1
Tag2
Tag3
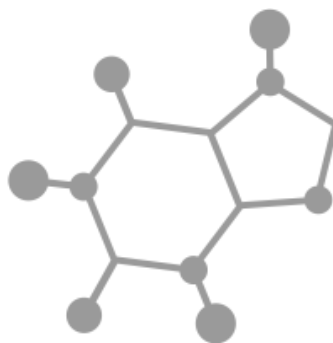
DEL data

Why Software Engineering? 🛠️

Good models need good code for real-world impact.

Programming languages: Python

Just like language, practice matters, its usage shapes your thinking.

- Require consistent practice to achieve fluency and proficiency.
- Learning one language will likely enable you to learn other languages faster.
- Will shape how you frame and solve problems.

Python?

- Fast (really!, but could be faster)
- Readable
- Well supported

PEP 8: Python Style Guide

PEP 8 is the Foundation for Readable Python.

PEP 8 is the widely adopted style guide for Python, offering recommendations on code layout, naming conventions, and overall code structure for clarity.
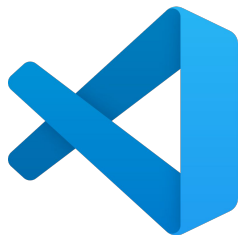
The Google Python Style Guide builds upon PEP 8 and provides stricter, more specific guidelines, particularly beneficial for larger projects and team collaboration.

https://peps.python.org/pep-0008/

https://google.github.io/styleguide/pyguide.html

In practice: Use An "Intelligent" Dev Environment

Code editor

Visual Studio Code

Pycharm

Auto-formatter, linter

https://github.com/psf/black

https://github.com/astral-sh/ruff

* Students can get github copilot for free

In practice: Use An "Intelligent" Dev Environment

Which GenAI/LLM to use?

Software engineering
SWE-bench verified

70.3%
with custom scaffold

62.3% — Claude 3.7 Sonnet

49.0% — Claude 3.5 Sonnet (new)

48.9% — OpenAI o1

49.3% — OpenAI o3-mini (high)

49.2% — DeepSeek R1

ACCURACY

https://www.swebench.com/

Introducing Style Guides

Style Guides Ensure Consistent Code Formatting.

Style guides provide a standardized set of rules for formatting code, promoting uniformity and reducing cognitive load when reading code.

```
1 def my_function(  input_value  ): # inconsistent spacing
2     if input_value>10: # no space around >
3         return input_value*2 # no space after return
4     else:
5         return input_value # inconsistent indentation
```

Coding mantras: The Zen of Python (PEP 20) 💡 🧘

```
 1 Beautiful is better than ugly.
 2 Explicit is better than implicit.
 3 Simple is better than complex.
 4 Complex is better than complicated.
 5 Flat is better than nested.
 6 Sparse is better than dense.
 7 Readability counts.
 8 Special cases aren't special enough to break the rules.
 9 Although practicality beats purity.
10 Errors should never pass silently.
11 Unless explicitly silenced.
12 In the face of ambiguity, refuse the temptation to guess.
13 There should be one-- and preferably only one --obvious way to do it.
14 Although that way may not be obvious at first unless you're Dutch.
15 Now is better than never.
16 Although never is often better than *right* now.
17 If the implementation is hard to explain, it's a bad idea.
18 If the implementation is easy to explain, it may be a good idea.
```

Code Readability Matters

Maintainable Code Enables Collaboration and Future Work.

🚧

```python
1 def process_d(d):
2   results=[]
3   for i in range(len(d)):
4     if d[i]>0:results.append(d[i]*2)
5   return results
```

Code Readability Matters

Maintainable Code Enables Collaboration and Future Work.

✨

```python
1 def process_positive_data(data_points):
2   """Processes a list of data points, keeping only positive
values and doubling them."""
3   processed_results = []
4   for point in data_points:
5     if point > 0:
6       processed_results.append(point * 2)
7   return processed_results
```

Readability: Starts with Naming

Clear Names Enhance Code Understanding

```
1 a = 10
2 temp = get_data()
3 flag = True
4
5 class Proc:
6     pass
7
8 class handler:
9     pass
```
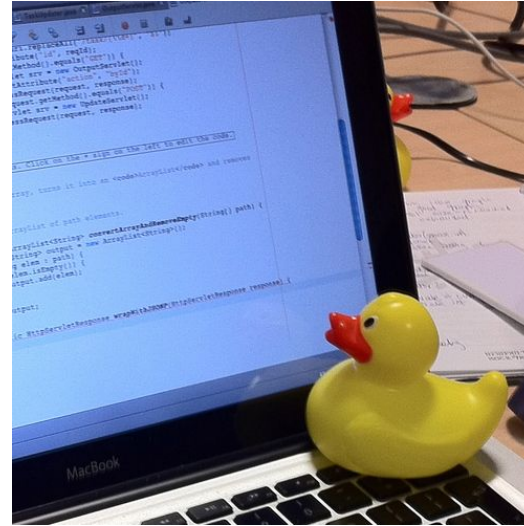
```
1 image_width = 10
2 user_data = get_data()
3 is_data_valid = True
4
5 class ImageProcessor:
6     pass
7
8 class UserDataHandler:
9     pass
```

- Functions: verb_do
- Variables: two_three_words
- Classes: CapitalizedWords

Pair-coding / Rubber duck programming

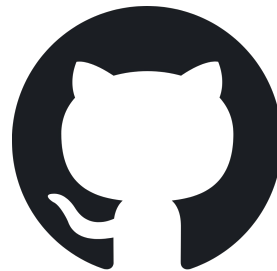Working in a social setting can help you to communicate thoughts and put you in a different attitude.

- https://en.wikipedia.org/wiki/Rubber_duck_debugging
- https://en.wikipedia.org/wiki/Pair_programming
- "[Strengthening the Case for Pair Programming](#)"
- https://pds.blog.parliament.uk/2017/03/29/pair-writing/

Software + Compute

≥ 3.11

github

Jupyter notebook
(online/offline) +
Google Colab (free)

Panorama

Exploratory
Data Analysis
With Molecules
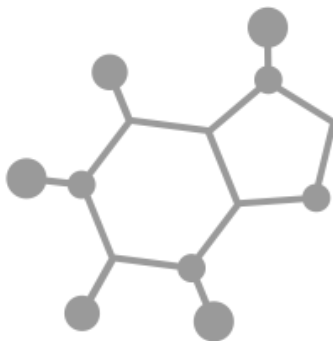

BB2
BB1
BB3
Tag1
Tag2
Tag3
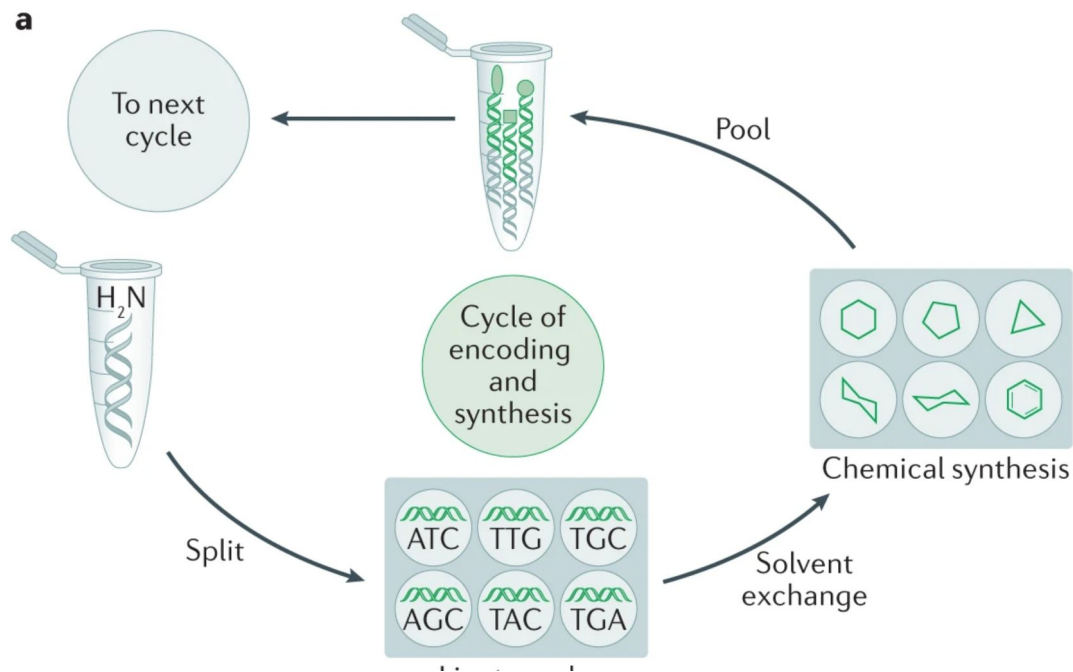
Coding in
'2025

DEL data

Panorama

Exploratory
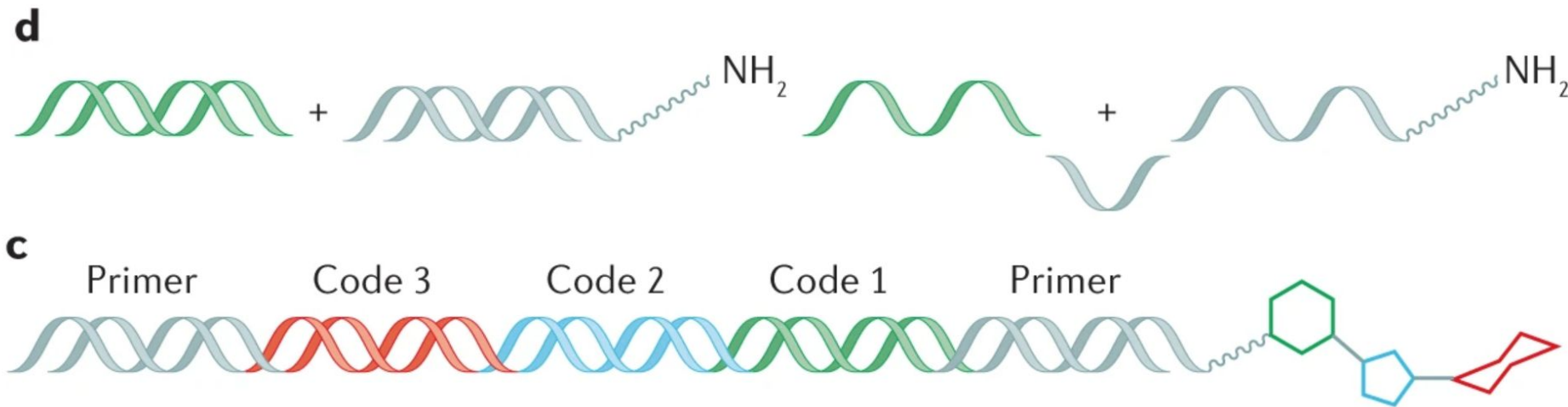Data Analysis
With Molecules

Coding in
'2025

DEL data

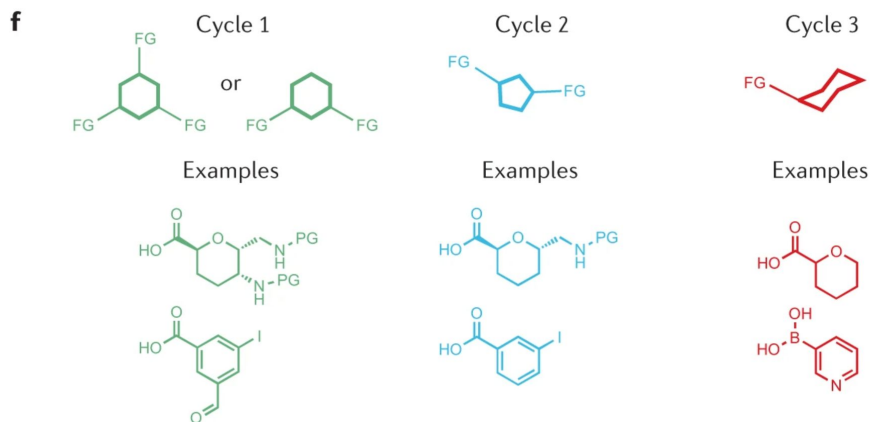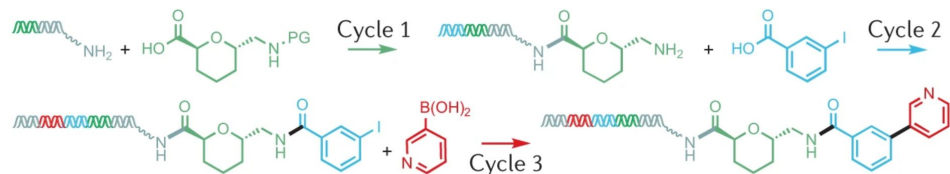# Unlocking Chemical Space with DNA-Encoded Libraries (DELs)

The Power of Combinatorial Synthesis and DNA Barcode

DELs utilize a "split-and-pool" synthesis strategy

Cyclic Synthesis and Chemical Diversity

DELs utilize a "split-and-pool" synthesis strategy

DNA-encoded chemical libraries, *Nat Rev Methods Primers* '22, 10.1038/s43586-021-00084-5

Thanks!

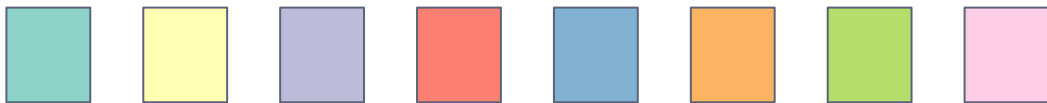# Questions?

Calling in for backup (slides)

For color palette

Dark2

Set3

Cividis (continuous)

PiYG (divergent)