Open-Science laboratory automation for AI-accelerated materials research and optimization

2024 ML4MS Workshop

Brenden Pelkie

May 13, 2024

CHEMICAL ENGINEERING
UNIVERSITY of WASHINGTON

Outline

1. Motivation: ML-guided accelerated experimentation
2. Autonomous experimental planning
3. Jubilee platform for experimental automation
4. Implementing an autonomous experiment

Intro: How do we design the best sunscreen?

Design objectives:

- Blocks UV radiation to prevent skin damage
- Doesn’t wash away
- Feels and smells nice

Parameter space includes

- Active ingredient(s)
- Emulsifier
- Preservative
- Fragrances

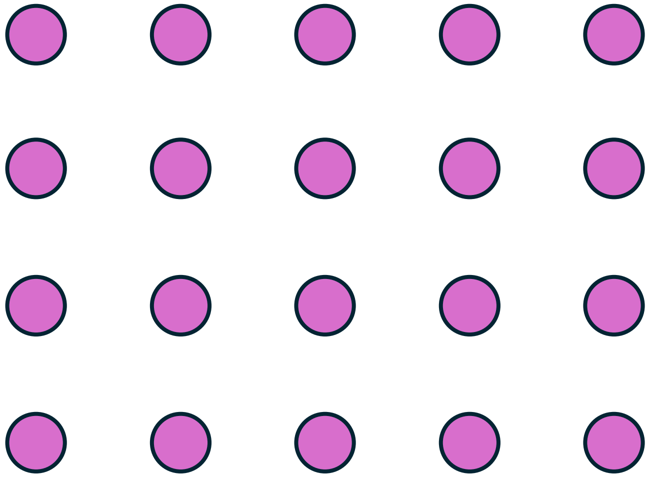
https://cdn.thewirecutter.com/wp-content/media/2023/06/facesunscreen-2048px-01008-2x1-1.jpg?auto=webp&quality=75&crop=1.91:1&width=1200

Traditional design of experiments

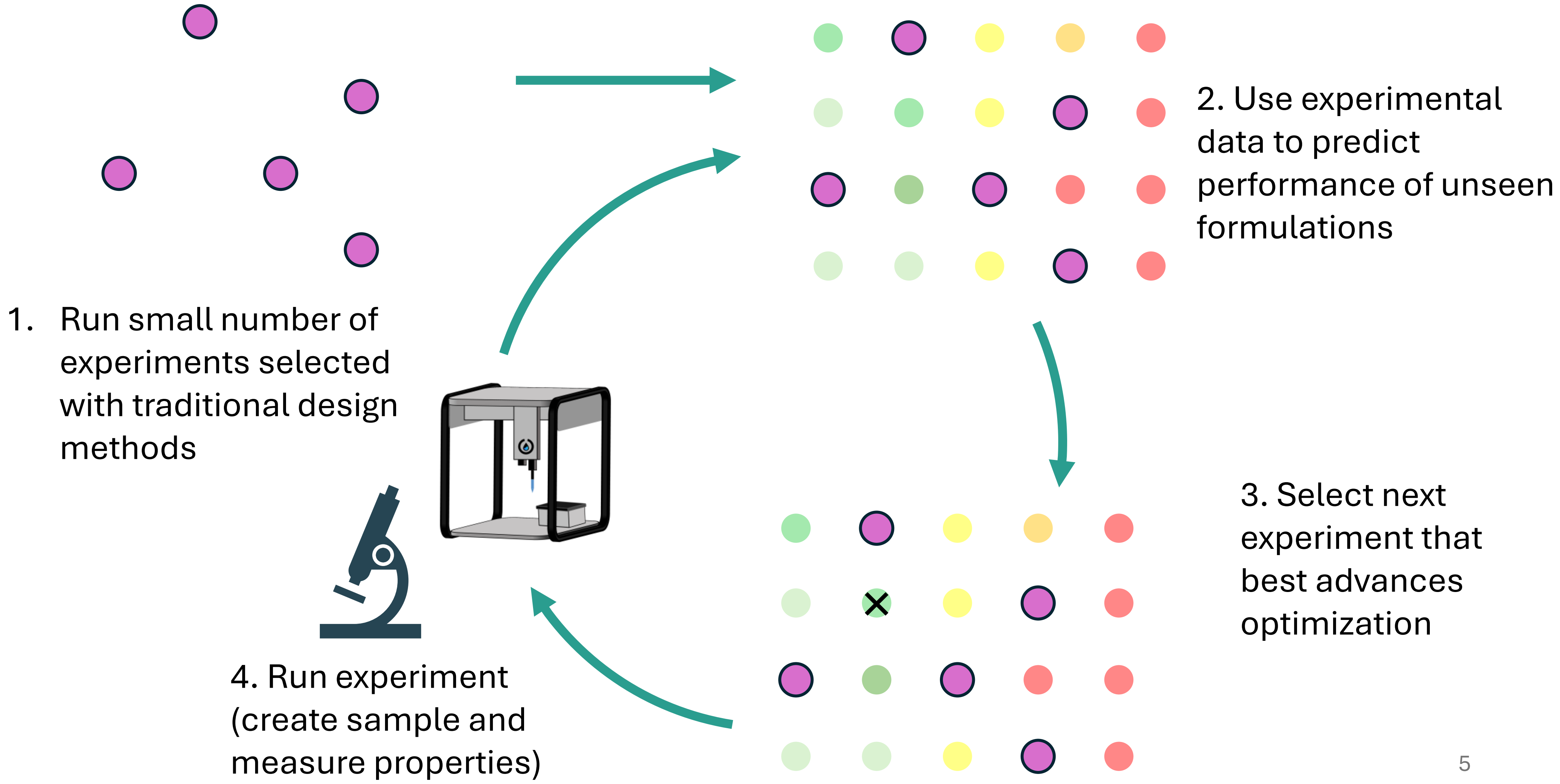
1. Pre-define experiments to run
    - Grid search, space filling design, random…
2. Run selected experiments

Issues:

- Spend a lot of resources on bad formulations
- Don’t optimize best results



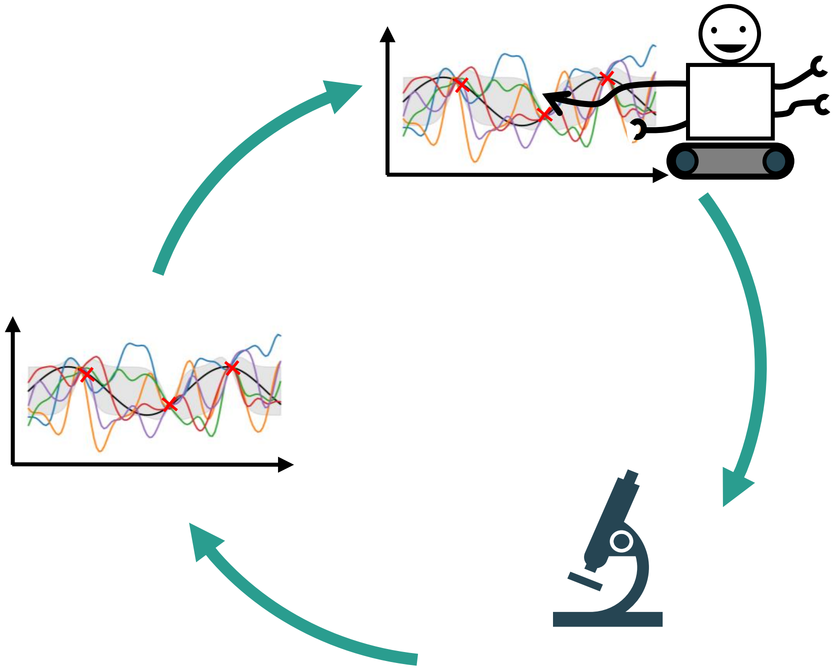
Adapt experiment plan on the fly



Two components for autonomous experiments

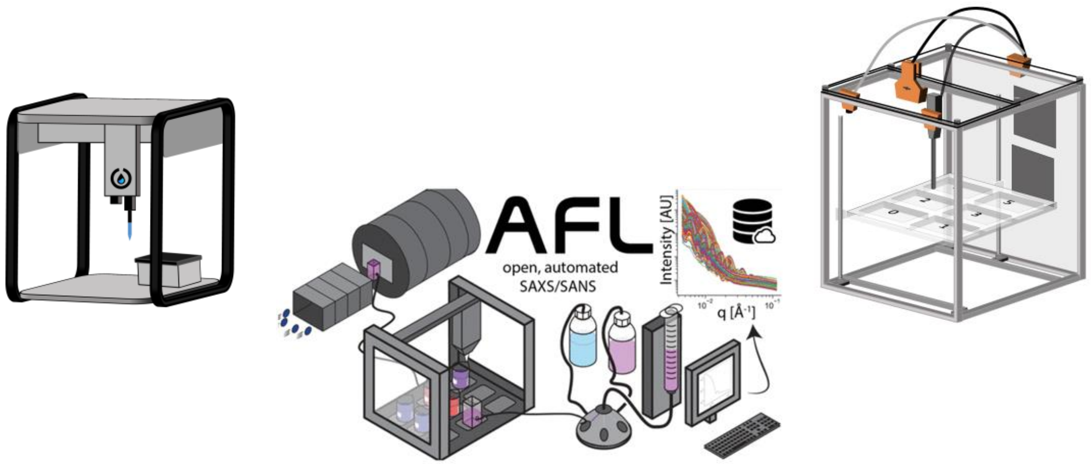
Autonomous experimental design

- ML-guided experimental planning to optimize towards a target
- Adapt experiments on the fly



Automated experimental execution

- Gold standard: No human intervention required after setup
- Automating everything is hard

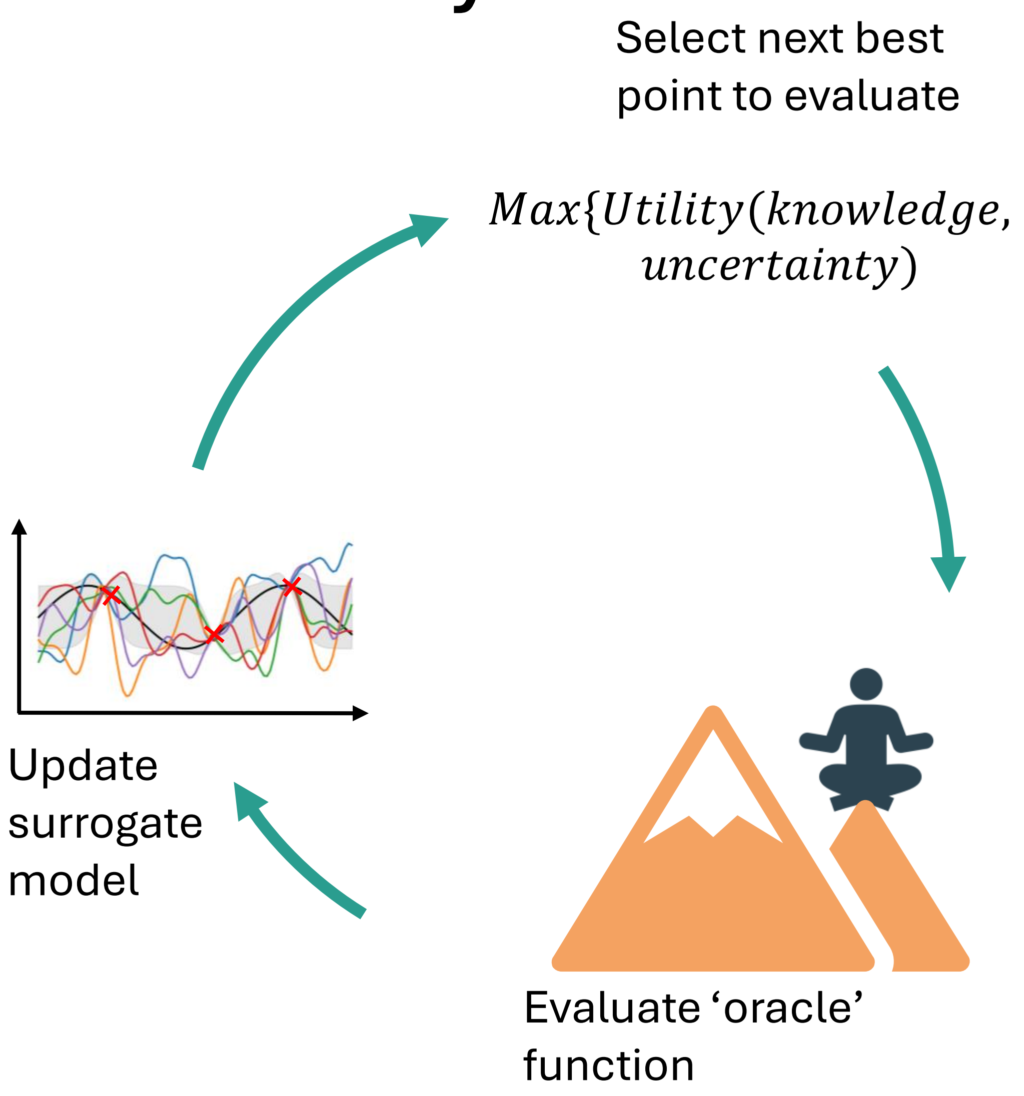


- Politi, Maria, et al. "A high-throughput workflow for the synthesis of CdSe nanocrystals using a sonochemical materials acceleration platform." *Digital Discovery* 2.4 (2023): 1042-1057.
- Beaucage, Peter A., and Tyler B. Martin. "The Autonomous Formulation Laboratory: An Open Liquid Handling Platform for Formulation Discovery Using X-ray and Neutron Scattering." *Chemistry of Materials* 35.3 (2023): 846-852.

How to select experiments on the fly?

Common Approach: Bayesian Optimization

- Relies on a surrogate ML model to select most useful points for experimentation
- Well suited to optimize expensive black-box functions

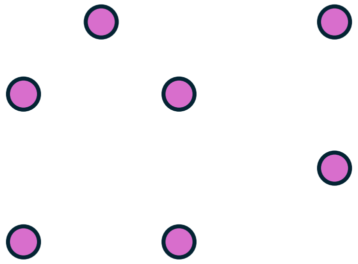


Experiment selection with Bayesian optimization

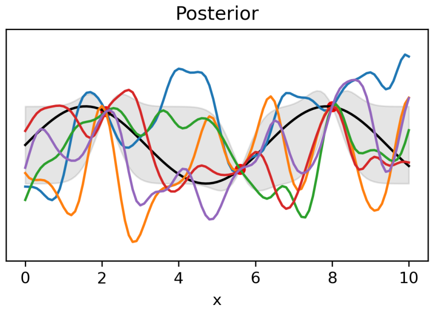
Bayesian optimization
- condition predictions about system on past observations

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

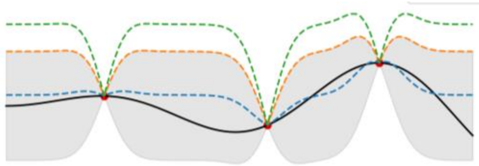
1. Collect initial observations of parameter – outcome mappings



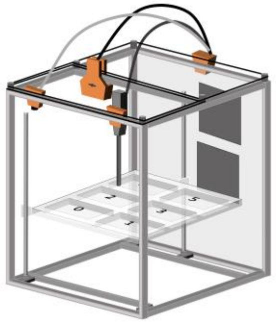
3. Fit surrogate model to observations



3. Maximize acquisition function over posterior distribution to select next sample point



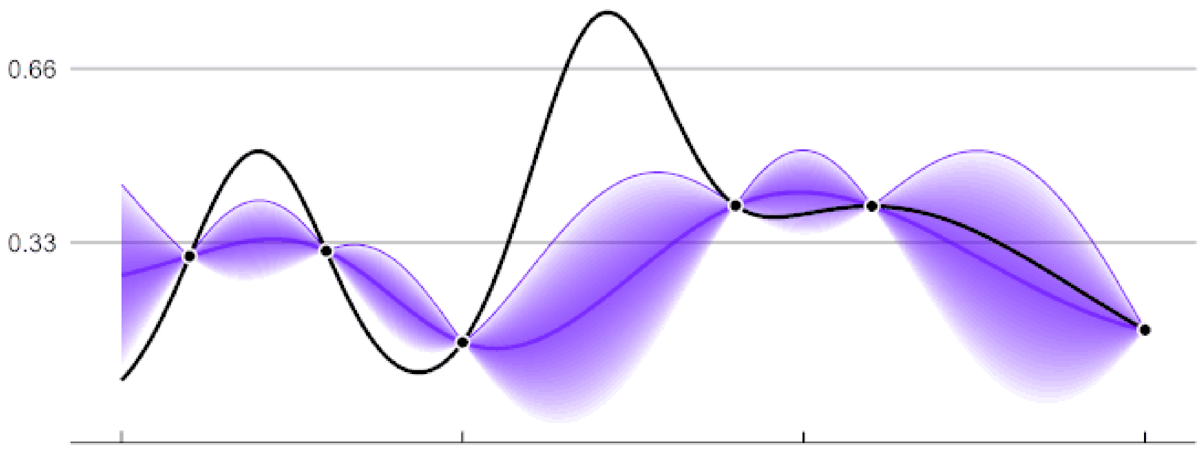
3. Execute new experiment and repeat



Gaussian Process Regression

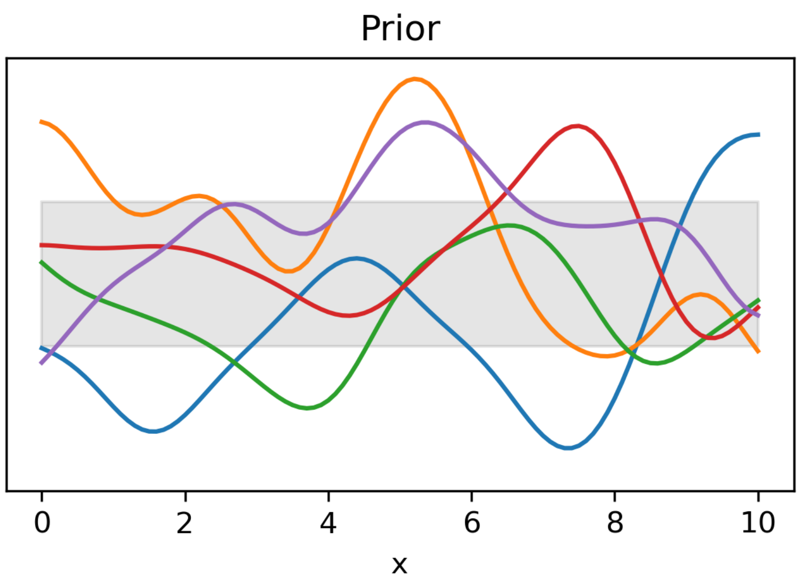
Non-parametric supervised machine learning model

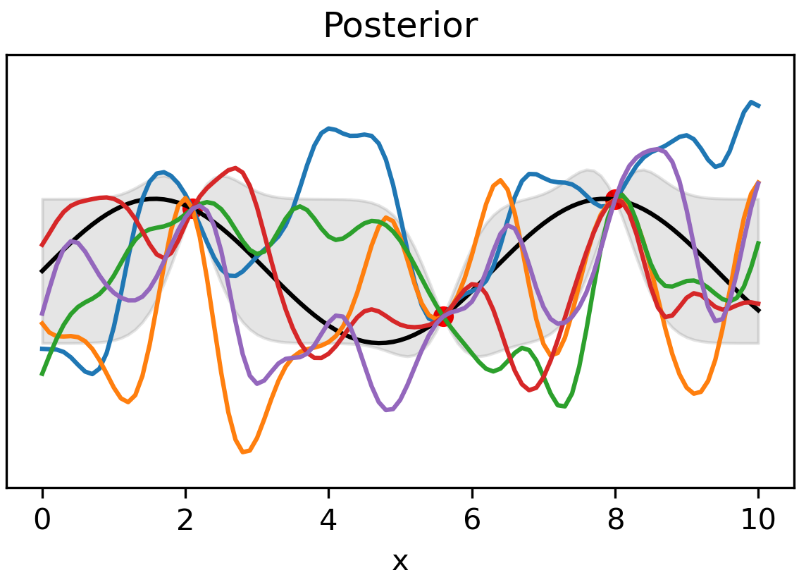
Gaussian process – Generalization of gaussian distribution to functions



https://upload.wikimedia.org/wikipedia/commons/0/02/GpParBayesAnimationSmall.gif

Gaussian process regression intuition





1. Prior is an infinite set of functions determined by kernel (covariance) function

2. 'Fit' by selecting functions from prior that agree with data

3. Make predictions by evaluating mean, variance of posterior

GPR Kernels

Appropriate selection is important consideration

Determines 'basis set' of functions to fit from

- Radial basis function kernel:

$$k(x_i, x_j) = \exp\left(-\frac{d(x_i, x_j)^2}{2l^2}\right)$$

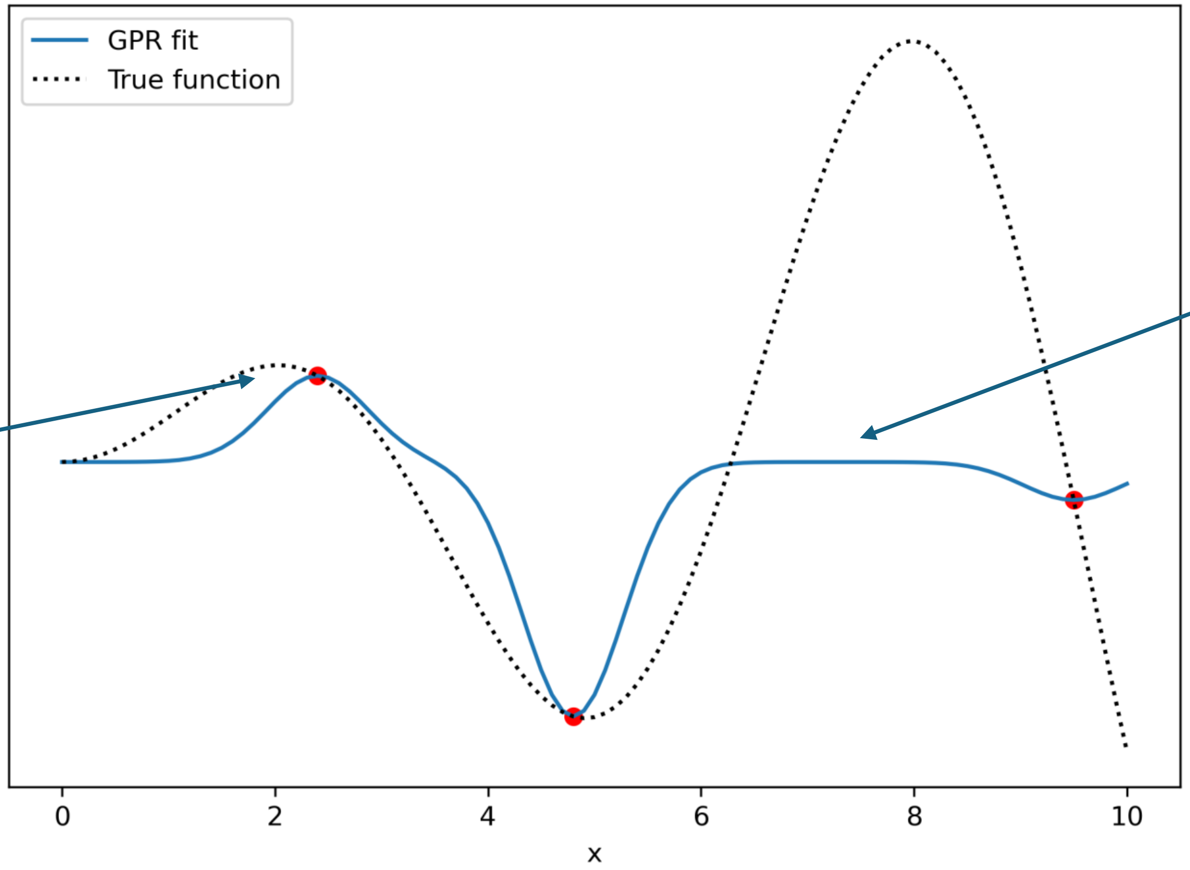
- Matérn kernel: Generalization of RBF with smoothness parameter

$$k(x_i, x_j) = \frac{1}{\Gamma(\nu)2^{\nu-1}}\left(\frac{\sqrt{2\nu}}{l}d(x_i, x_j)\right)^{\nu} K_{\nu}\left(\frac{\sqrt{2\nu}}{l}d(x_i, x_j)\right)$$

Acquisition functions for Bayesian optimization

What's the most useful experiment to run next?

- Naïve approach: $argmax(surrogate\ model)$



Local optima

Unexplored regions

Acquisition functions for Bayesian optimization

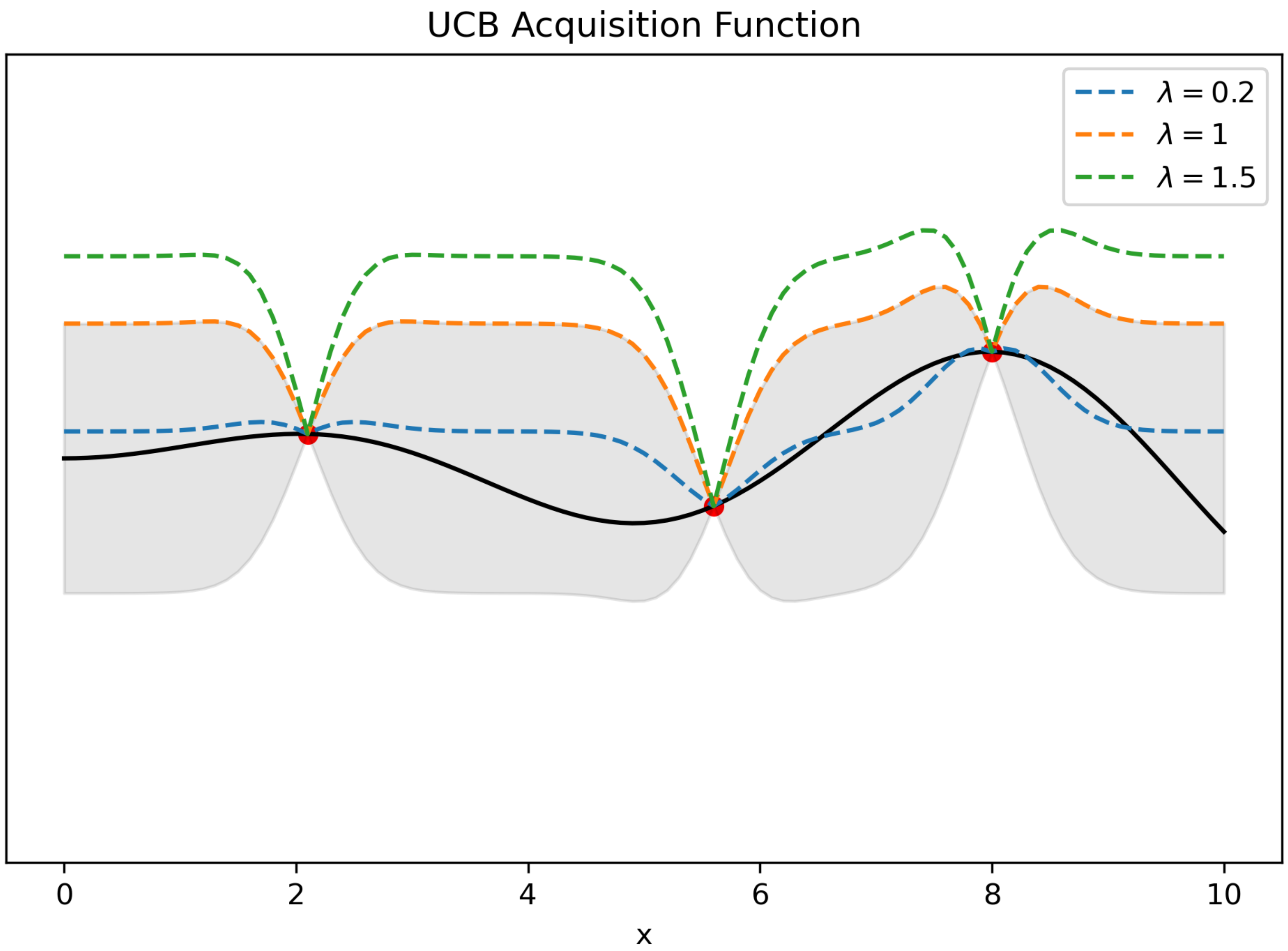
- Acquisition function considers explore-exploit tradeoff

Explore non-tested areas of parameter space

Exploit parameter values with promising performance

- Upper confidence bound: $a(x) = \mu(x) + \lambda\sigma(x)$
- Expected Improvement/Probability of Improvement also common

Upper confidence bound acquisition function



Implementing Bayesian optimization

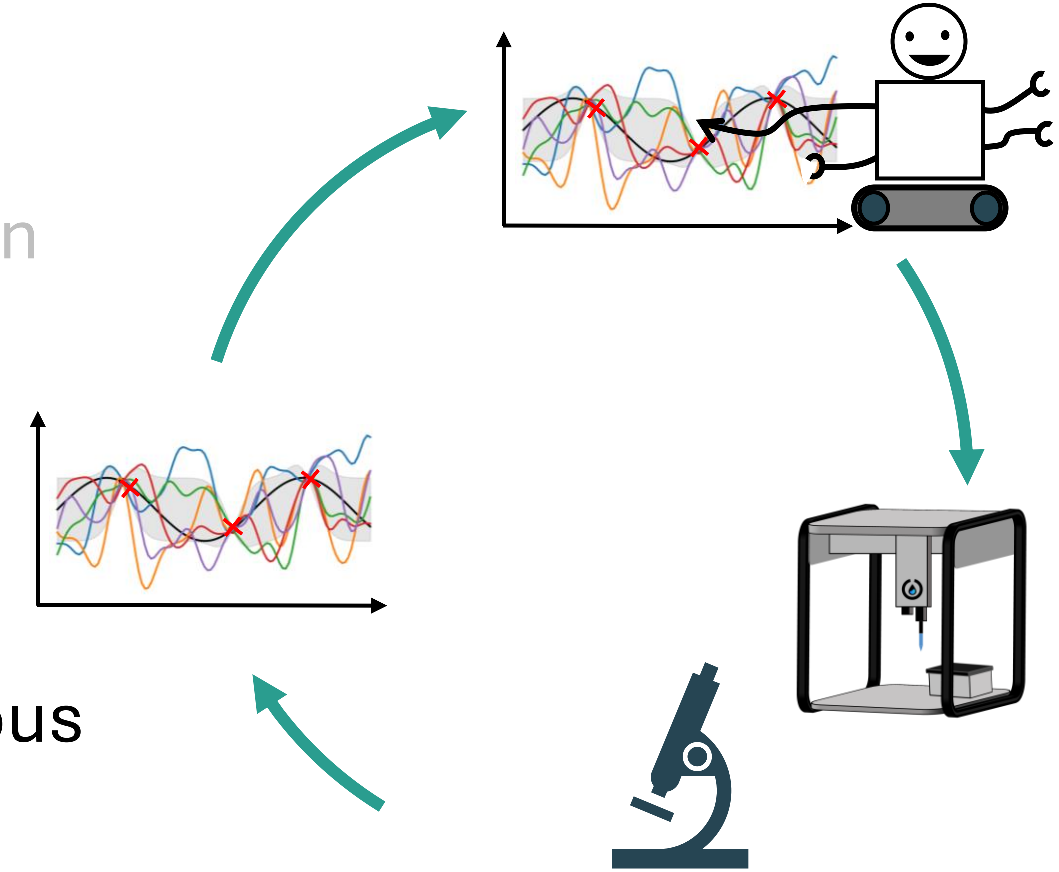
Many well supported implementations:

- BoTorch: Built on PyTorch. ‘Ikea furniture’ approach
- Ax: Built on BoTorch. No assembly required
- Bayesian Backend (BayBe): Chemistry/materials focused

- Many other possible ML algorithms
- Goal: Show integration to experiments

Outline

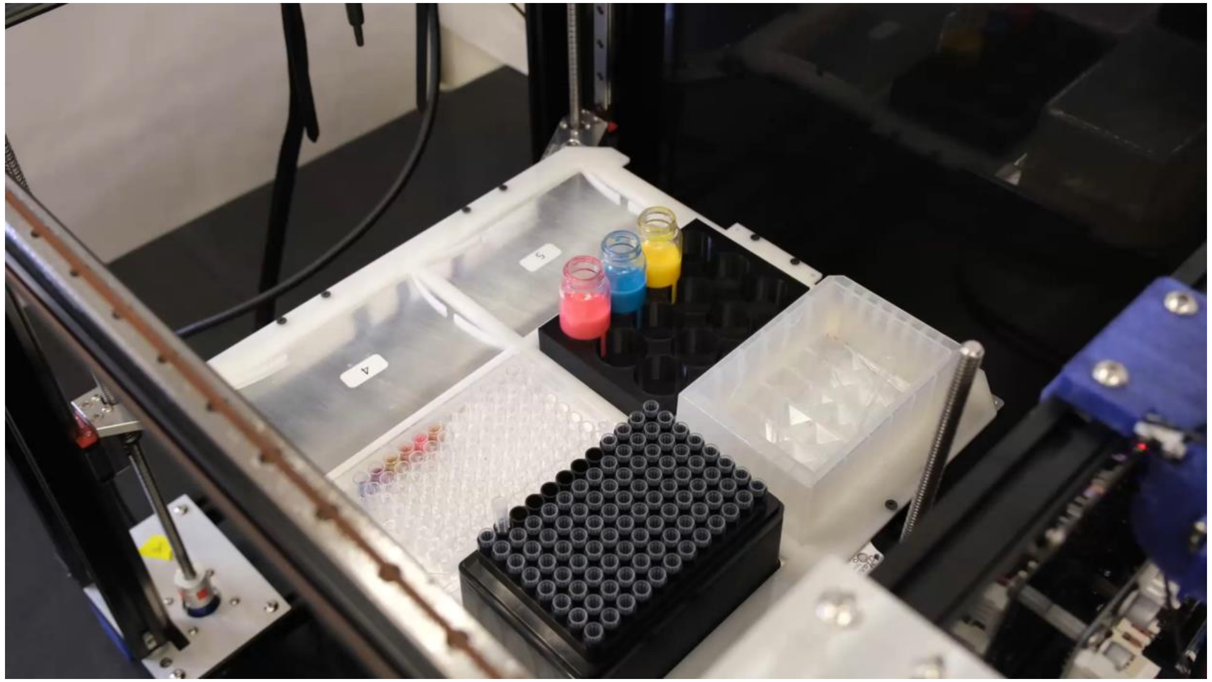
1. Motivation: ML-guided accelerated experimentation
2. Autonomous experimental planning
3. Jubilee platform for experimental automation
4. Implementing an autonomous experiment



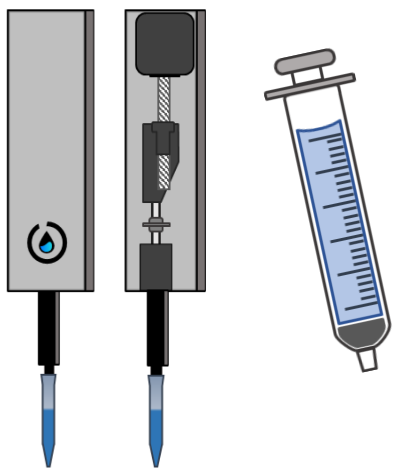
Jubilee: Open, flexible automation

Jubilee is a tool-changing 3D motion platform

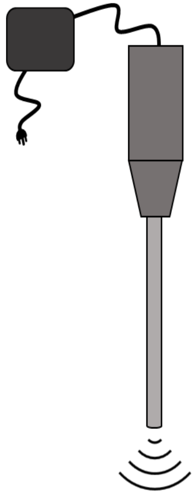
- Tool changing: can incorporate multiple tasks into a workflow
  - Synthesis + Characterization on one platform
- Open hardware: users build from a kit and have full control over platform
- Documented tool interface enables new tool integration



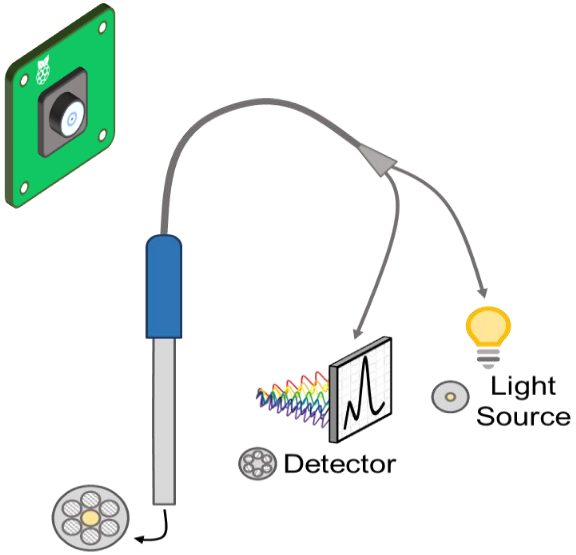
From sample formulation to fabrication tools



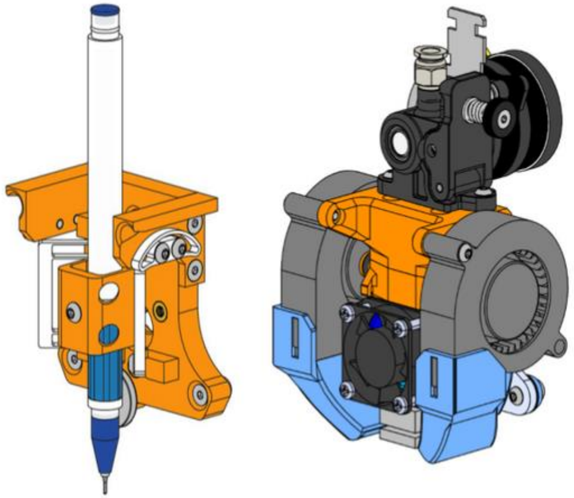
Sample formulation



Sample Processing



Sample Characterization

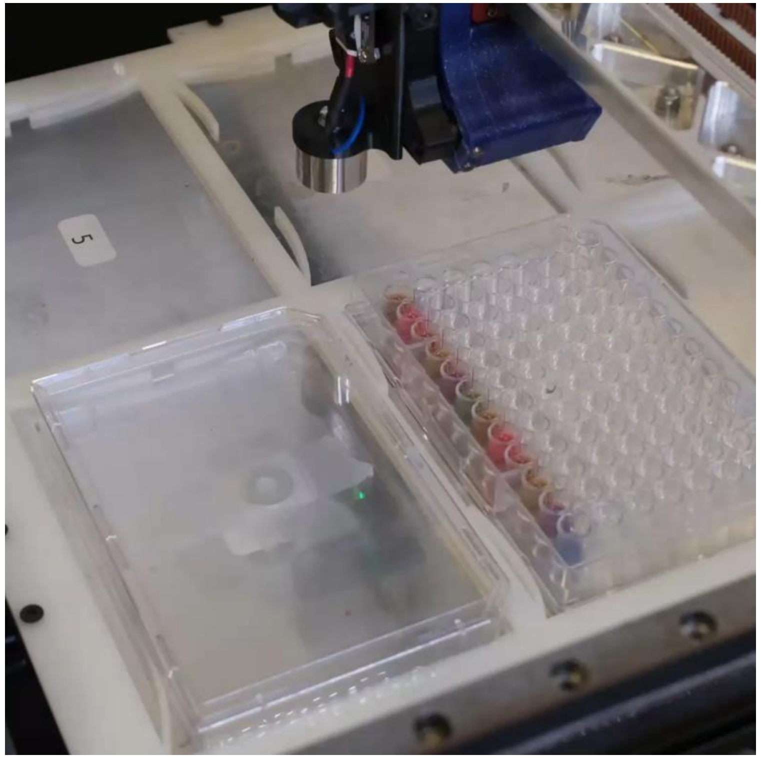


Fabrication and more

New tools for custom applications

Haptic-vibration mix plate

Electromagnet for lidding/de-lidding labware



Science-focused control software

Jubilee is controlled by ‘g-code’: Machine control language used in 3D printing

- Example: G1 X50 : Move X-axis to 50 mm
- Requires knowing positions
- Very easy to crash
- http://localhost:192.168.1.2

A simple Python library for Jubilee

```python
jubilee = Jub.Machine(address='192.168.1.2')
```

Python

```python
deck = jubilee.load_deck('lab_automation_deck.json')
tiprack = jubilee.load_labware('opentrons_96_tiprack_300ul.json', 0)
samples = jubilee.load_labware('corning_96_wellplate_360ul_flat.json', 2)
stocks = jubilee.load_labware('20mlscintillation_12_wellplate_18000ul.json', 3)
trash = jubilee.load_labware('agilent_1_reservoir_290ml.json', 1)
```

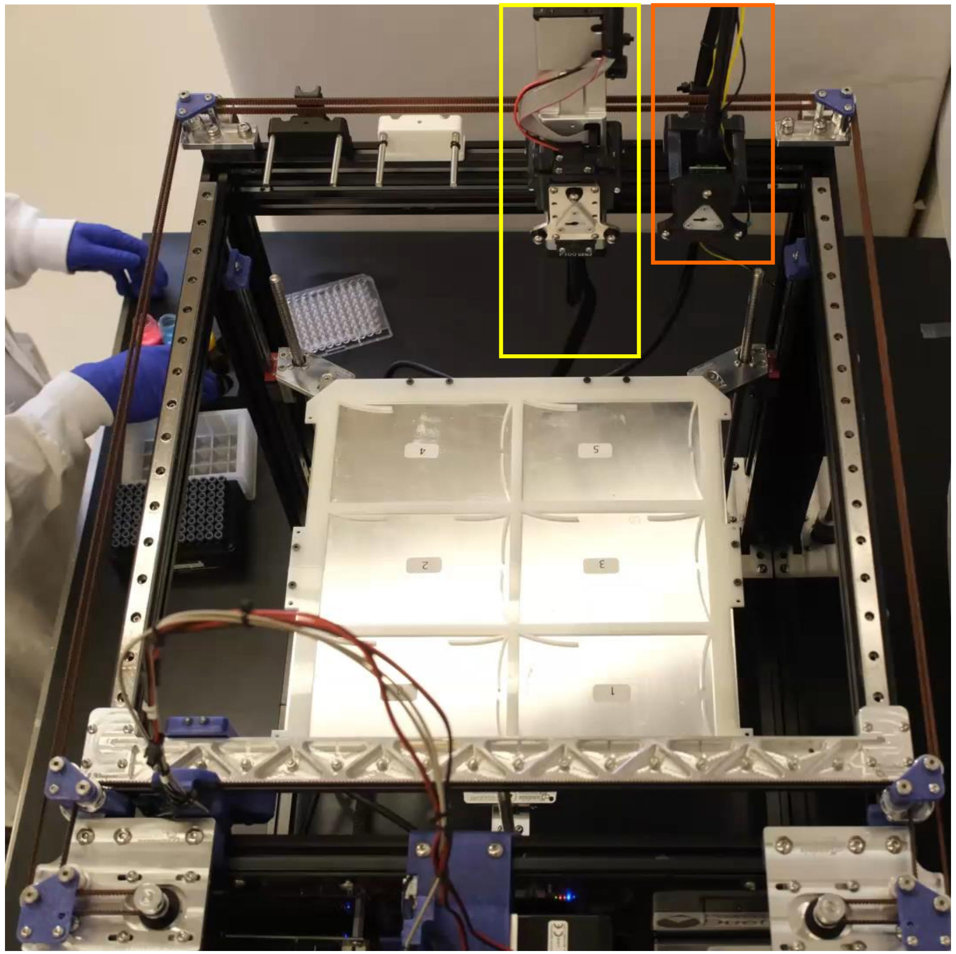
Python

```python
#configure pipette
P300 = Pipette.Pipette.from_config(1, 'Pipette', 'P300_config.json')
jubilee.load_tool(P300)
P300.add_tiprack(tiprack)
```

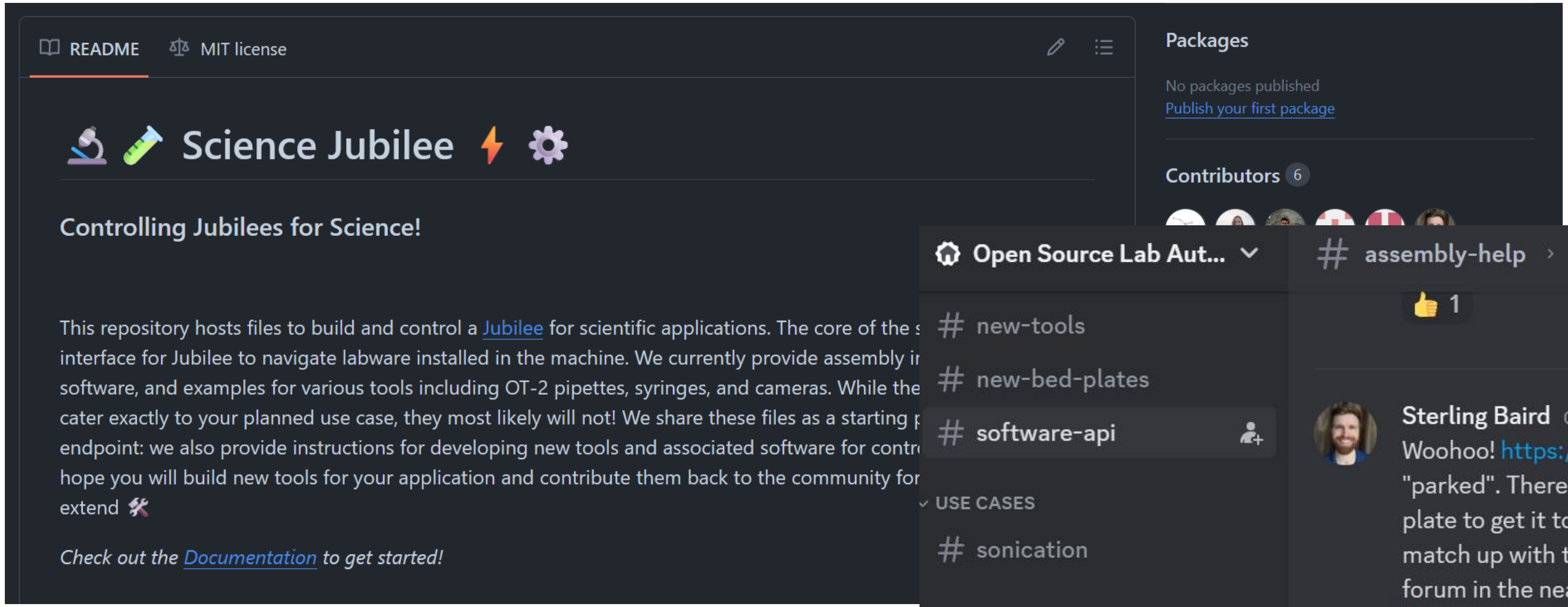
Python

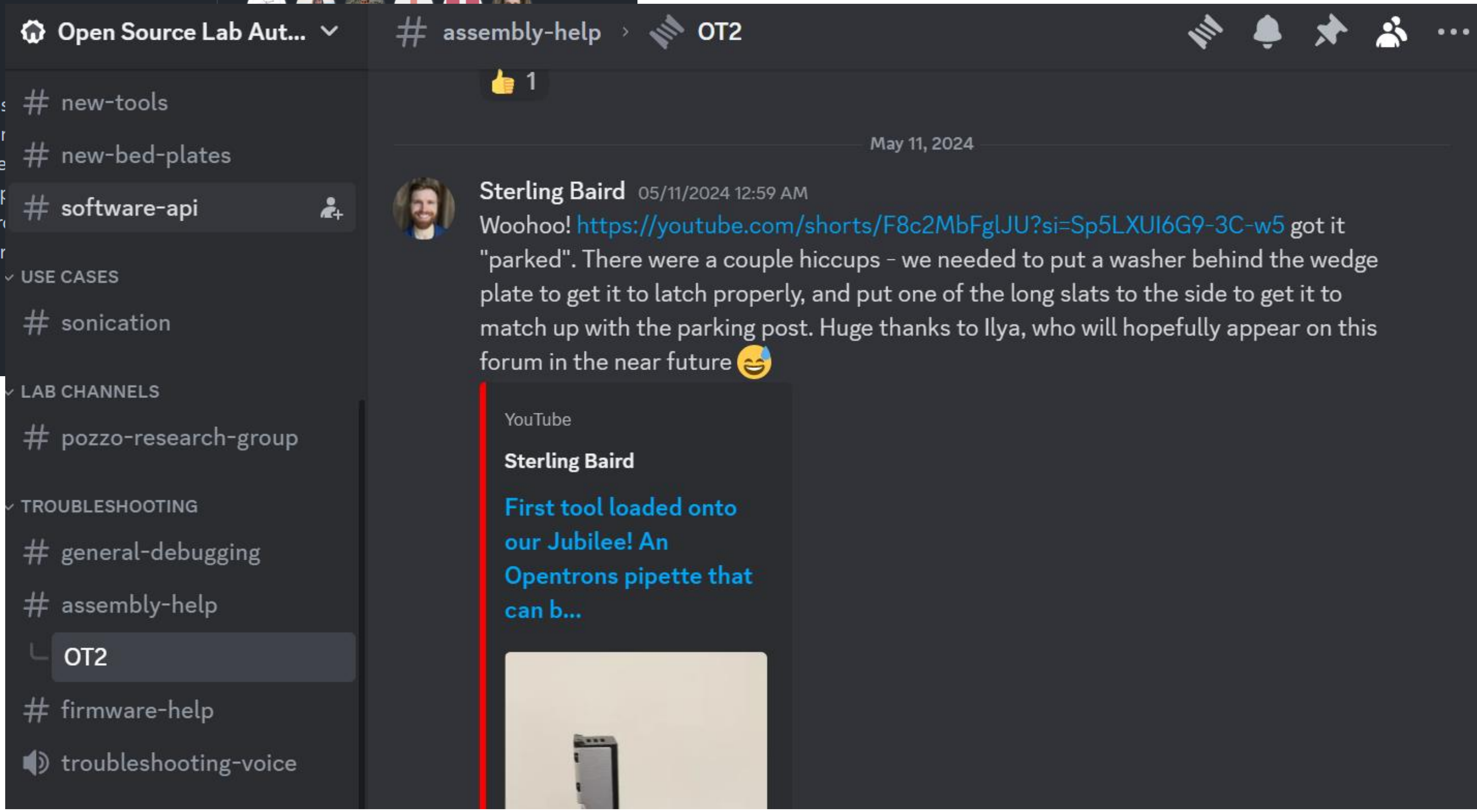
```python
#configure camera
Camera = WebCamera.Camera.from_config(0, 'Camera', 'WebCamera_config.json')
jubilee.load_tool(Camera)
```

Python



Jubilee community





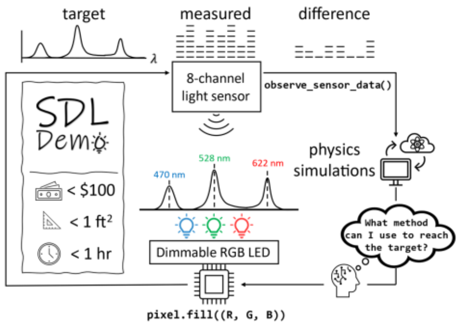
- Community contributions on GitHub
- Discord for support and development

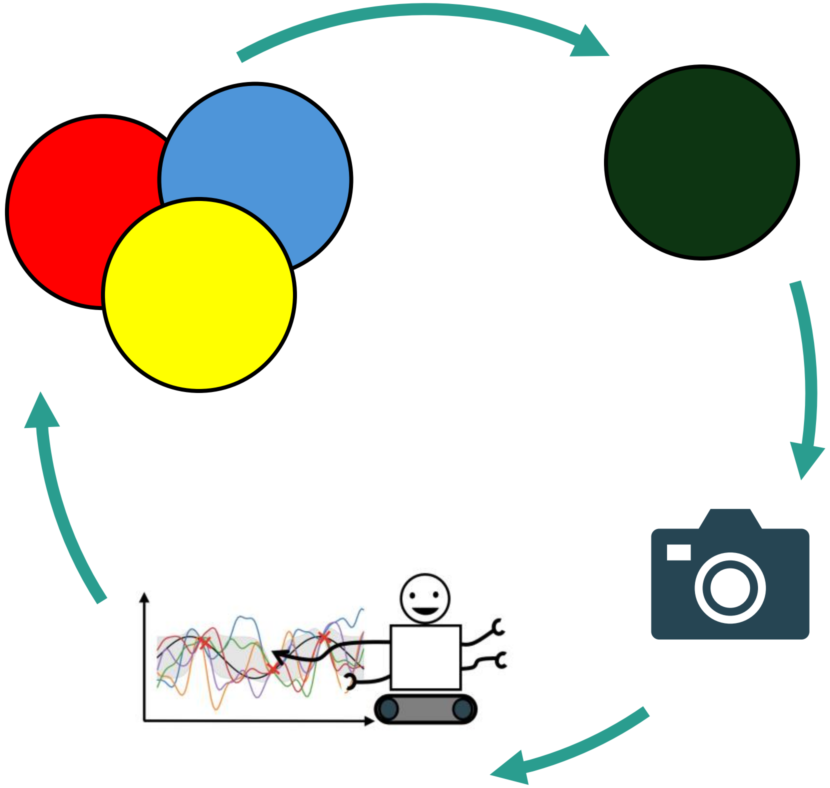
Platform demonstration with color mixing

Goal: Learn to make a target color from a selection of base colors

Great test case and demo of autonomous experimentation:

- Intuitive and understandable
- Closely matches ‘real science’ workflow requirements
- Tunable complexity to fit needs

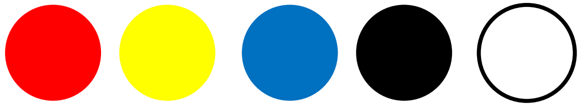




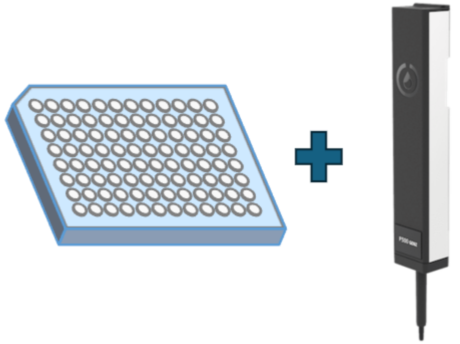
- https://github.com/sparks-baird/self-driving-lab-demo?tab=readme-ov-file
- https://sites.google.com/matterhorn.studio/sdl4kids/home

Our color mixing implementation

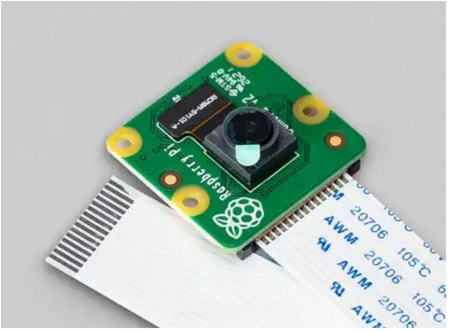
Parameter space:



Paint mixtures (with volume fraction constraint)
Target: User-selected RGB value



1. 'Synthesize' sample with pipette



2. 'characterize' sample with a Raspberry Pi camera

3. Extract RGB value of sample from picture

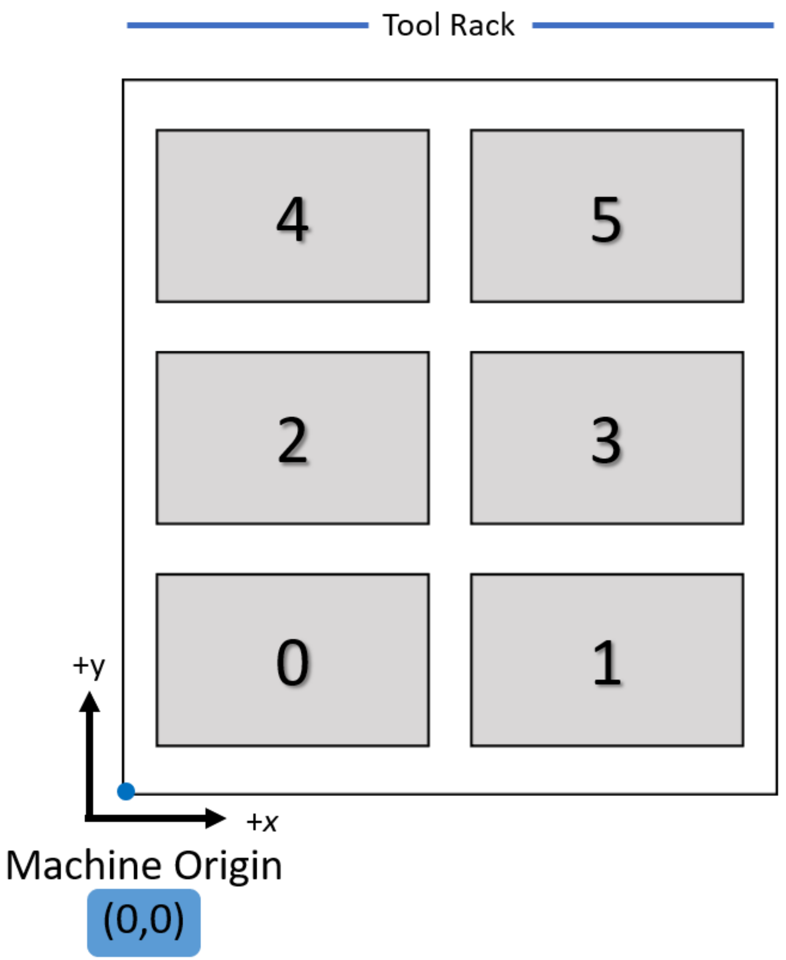
```
## Code from initialize_model
kernel = MaternKernel(nu = self.nu)
gp_model = SingleTaskGP(self.data_utils(normalized_x), self.data_utils(y_data),
outcome_transform=Standardize(m=1), covar_module=kernel).to(normalized_x)

mll = ExactMarginalLogLikelihood(gp_model.likelihood, gp_model)

fit_gpytorch_mll(mll)
```

4. Euclidean distance metric for RGB distance, Select next sample using BO implemented in BoTorch with some glue code

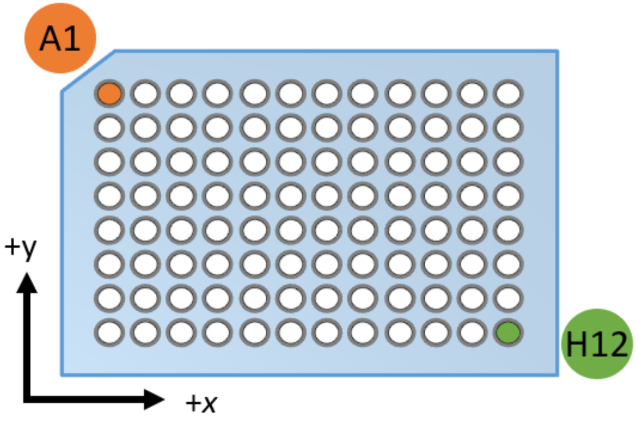
Images:
https://shop.opentrons.com/pipettes/
https://www.raspberrypi.com/products/camera-module-v2/

Automation labware

Deck Top View



Wellplate Top View

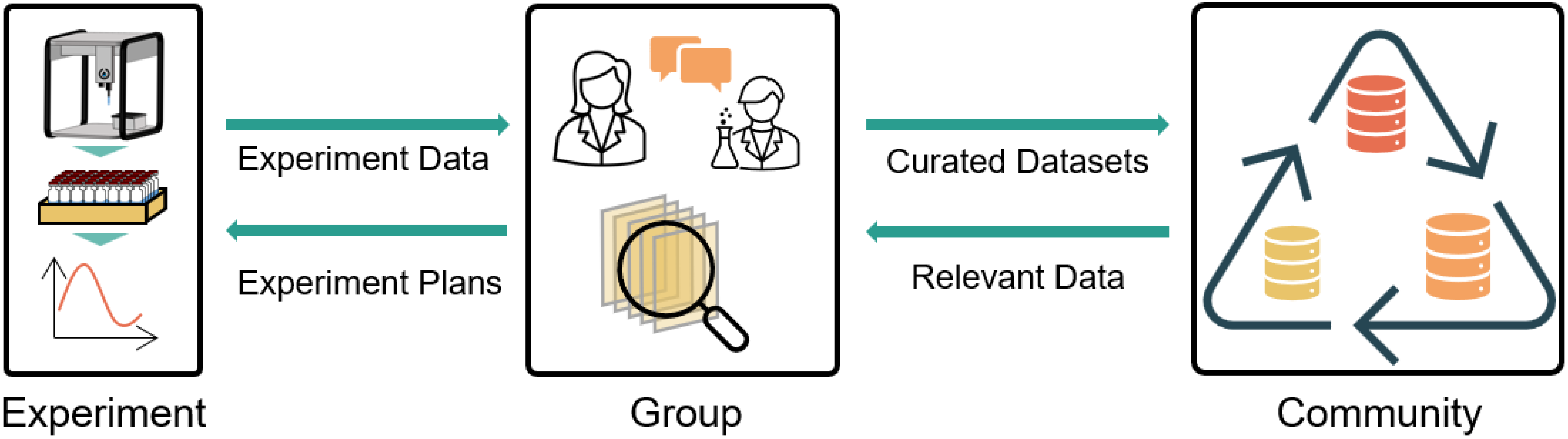


Demonstration Component

- Bayesian Optimization setup
- Jubilee control
- Full color mixing demo

Data management for autonomous science

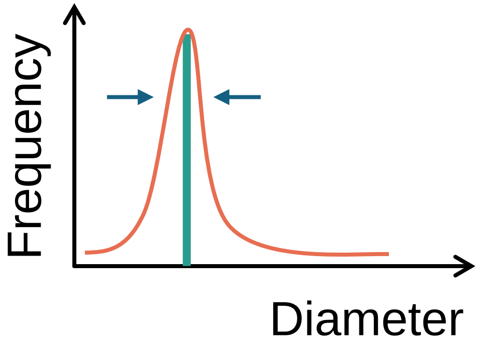
Automating experimentation is a great chance to fix how we do data: You're already building a new data pipeline, why not make it FAIR?

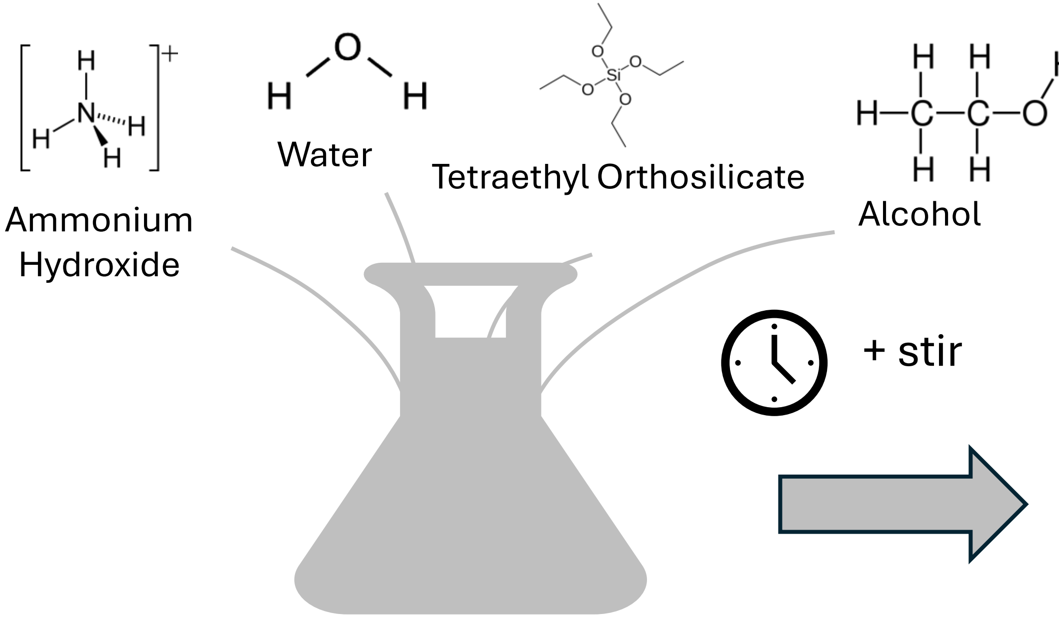


Pelkie, Brenden G., and Lilo D. Pozzo. "The laboratory of Babel: highlighting community needs for integrated materials data management." *Digital Discovery* 2.3 (2023): 544-556.

Autonomous nanoparticle synthesis

- Can we optimize nanoparticle morphology with an autonomous approach?





Closing thoughts

- Challenges in automating experiments
  - Difficult samples (volatile, air-sensitive, toxic)
  - Extreme conditions
  - Integrating external instruments and capabilities
- While challenging, autonomous experimentation is doable
  - Closes gap between predictions and AI applications to materials development
  - Many infrastructure options, Jubilee is one

Acknowledgements

Specific contributions:

- Maria Politi: Jubilee project, figures
- Blair Subbaraman, Sonya Vasquez, Sam Ferguson, Cecilia Abella - Jubilee

- Pozzo Research group

Funding:

- UW Clean Energy Institute
- UW MEM-C

CLEAN ENERGY INSTITUTE
UNIVERSITY of WASHINGTON

UW MEM·C