# AI/ML Discussion

Peter Beaucage

NIST NCNR

a self-driving *lab*(rador),
who has an altogether appropriate reaction to the idea of a dog or an AI running an experiment

# Who am I / why should you listen to me?

Staff scientist @ NCNR SANS,

beamline dev @ NSLS2 RSoXS,

formerly CHESS MatSci SAXS

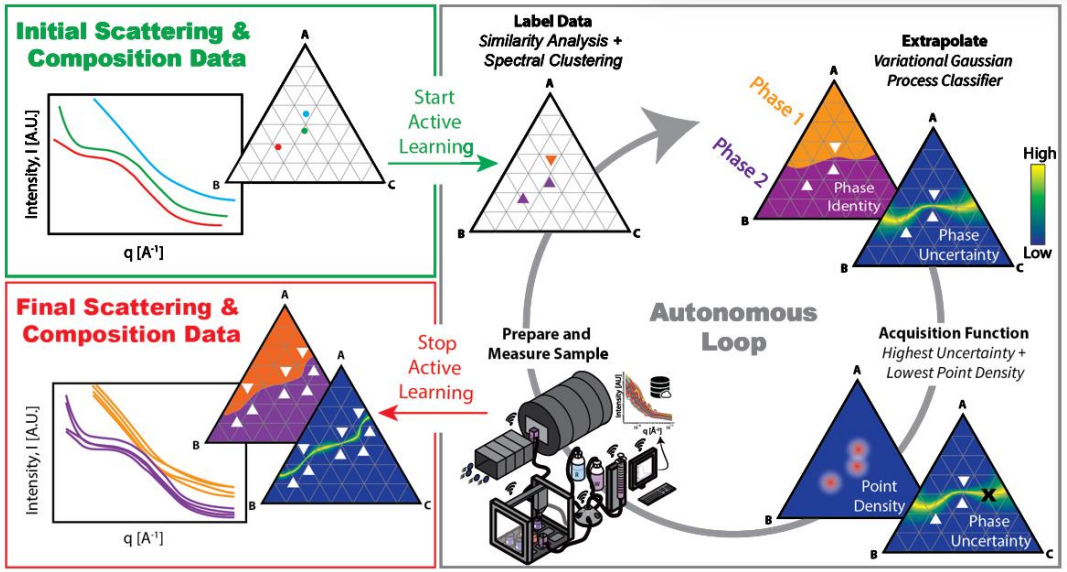
an eternity ago APS USAXS


Now run (w/ Tyler Martin) *NIST AFL,*

a nomadic/distributed, AI-driven self-driving lab for SAXS/SANS

~weeks each at NCNR, CHESS, APS x 2, NSLS2, ISIS, Diamond, SINQ

*and we have **thoughts***




Friday, 8:55am, 3F

# Why should you care about ML/AI/self-driving tech for SAS?

- Efficiency improvements,

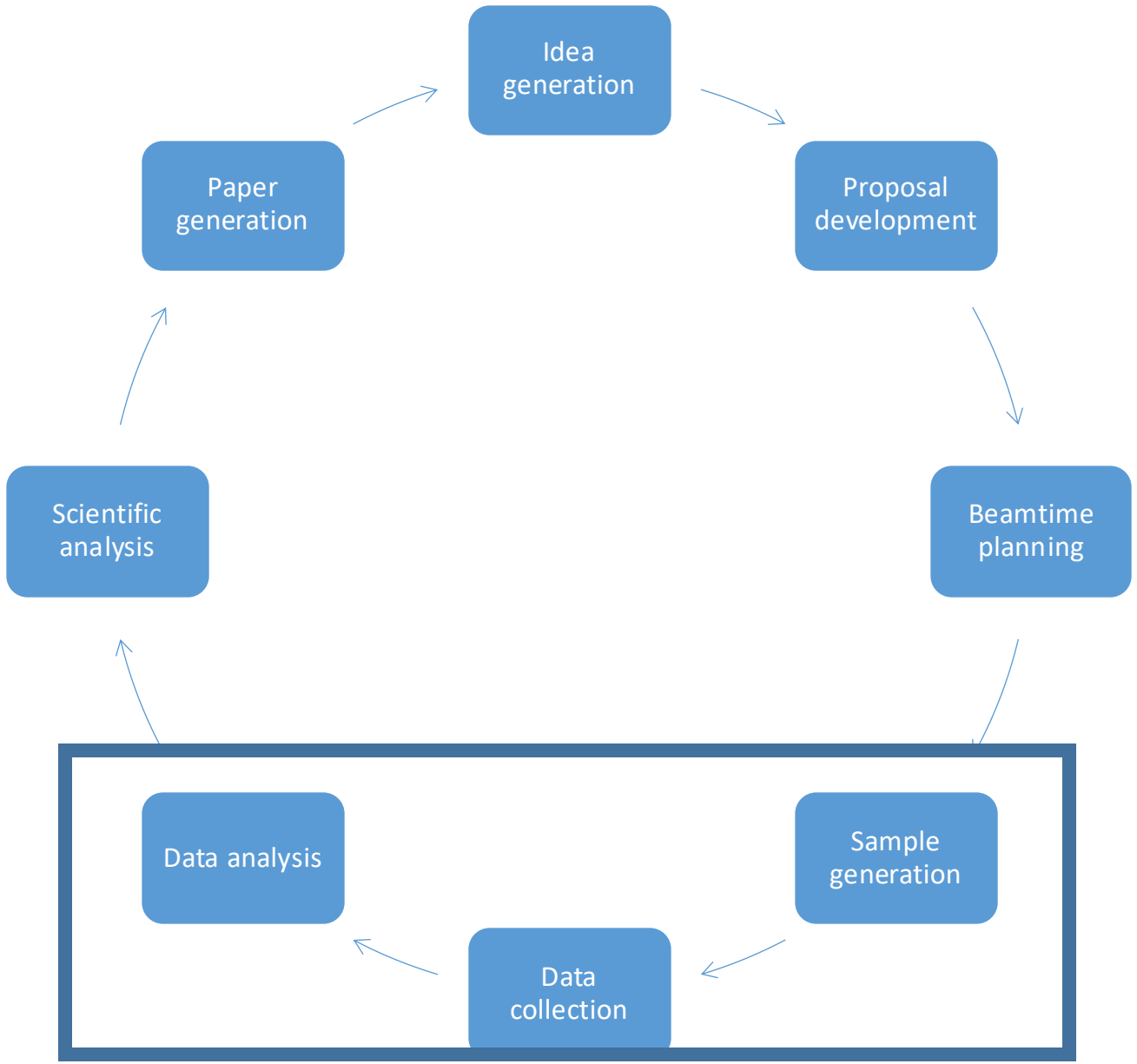
  which could also be phrased as data quality improvements

- Automation improves reproducibility and allows targeting rare phenomena

- Cooperative discovery between multiple techniques/facilities.


10.1021/acs.macromol.4c01410

# AI is ~~going to be~~ pervasive




- The most impactful AI applications will be outside the facility wall

these mostly demand good, reduced data (see earlier)

- AI moves *scary fast*, and facilities don't.

This is a feature! Let users or other facilities do/share hard work.

So, the question is:

- How can facilities support user-supplied or portable AI decision engines?

# Needs for AI decision support engines

1. Decision-ready data, through an accessible, documented API

2. Instrument control, through an accessible, documented API

# Decision-ready data

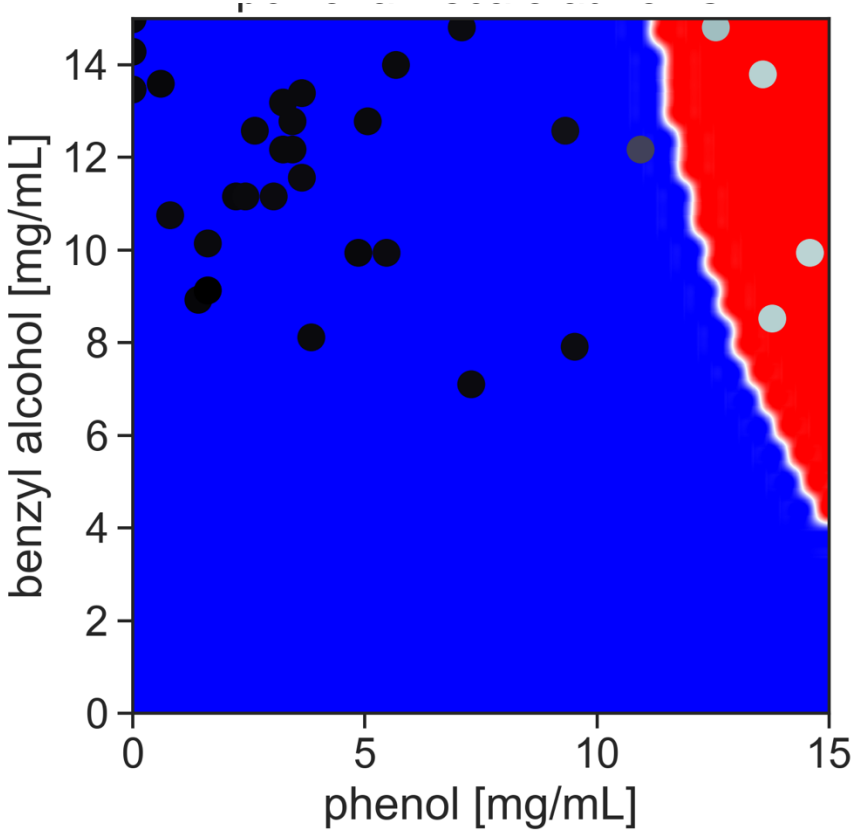
- For *most* cases, all the information needed for reduced/subtracted data can exist as the data are taken.

- Analyzing half-reduced data is the current state of the art, and …doesn't work well.
- Metadata are as important as the data. JSON adjective taxonomy would be useful.

Needs/practices:

- pipelining, user-friendly software
- standard descriptors of reduction 'unit ops'?

Bespoke GUI reduction is unnecessarily user-unfriendly, AND prevents future data-intensive work.



# Accessible instrument control

Who drives the experiment?

Existing software is almost always focused around the *instrument as the center of the universe*, but integrating AI into constrained software/hardware envs is hard.

We should more regularly support outside decision engines as the ‘center of the universe’ with the SAS instrument as a subordinate worker.

What endpoints are useful?

- change sample environment condition
- perform measurement (with parameters)
- change sample

How do we actually do it?

Facility IT security is a real concern.

Protocols are in principle shared (EPICS, Tango, spec, Bluesky, NICOS, …),

but enough instrument specifics to prevent perfect plug-n-play

# Accessible, documented API

- "The Magic Cable"
  CAT5e onto a "user-inaccessible" network
  SSH key to a jump node to another jump node
  "secret" instrument password
  instrument scientist's personal password

***this is terrible***

- Tell me I'm wrong: both of these APIs should be publicly accessible HTTP endpoints with rich authentication/permissions, routine security testing.

  *HTTP doesn't support high data rates* → Netflix, live TV, ...

  *HTTP isn't secure* → Cisco, Juniper SSL VPNs, all web apps

  *the public internet is an unnecessary risk* → Zero Trust says that your facility network is basically the public internet already.

- Documentation, past examples, libraries in various languages

# Ongoing efforts to watch


Tiled, secure streaming data access API deployed for *raw* data at NSLS2

Prefect and similar “workflow management” tools
Robust UI for data reduction on-the-fly and re-running

ORNL remote instrument API → EPICS PV read/write with rich authentication over HTTP on open networks

Other efforts to watch?

It’s unrealistic to expect one universal standard for this, but open, documented, and available without “magic cables” or hacking into a facility is going to enable the most transformative changes to how we conduct experiments.