Open Source

1) What is your favorite platform to share lab protocols or automated scripts? Why do you love it?
    - Google Drive
    - eLabs for integration and ease of use
    - GitHub is great for version tracking, backup, and security
    - Email, and we don’t love it
    - Opentron, GitHub, Lab book online for the whole lab
    - AskLena.com or AskRita.com *(refers to other lab members)*
    - GitHub. Area lacking for this type of sharing of metadata.

2) What do you dislike the most about open-source?
    - Lack of support
    - Tech bro cultures/competition
    - Sorting through different packages for good quality open source packages. Limited review which is also its strength
    - Open source can mean questionable quality. Having better certification or reviews can help with that.
    - Lacking organization, validation, and maintenance
    - Lack of dedicated tools that can solve one issue effectively, needs to always be built upon
    - Need more resources as not plug and play. Could be issues with open source as not as well validated. Support harder as not known software.

3) How can biofoundries improve sharing protocols?
    - Documentation of failed protocols so people don’t make the same mistakes (creating iterative versions of how a protocol is formed & sharing access to that w results)
    - Standardize protocol descriptions "ontologies"
    - Having a repository of protocols that is easily searchable, and having a fundamental naming and ranging conventions.
    - GitHub for sharing scripts. Making protocol/methods journals free access.
    - Public databases that can be easily accessed. Pushing journals to have more modern “methods sections” (ie: videos, GitHub links, etc). Reviews and certifications of published automation protocols.

- Standardization of nomenclature in protocols. Want ability to link analysis to protocol.
- Creating a universal language (having representation from communities that use standardization of these languages)

Training/Education

1) What 3 skills are most valuable in a new hire?
    - 1) Ability to deliver project 2) Listening 3) Fun and kind
    - 1) Experience 2) Curiosity to learn (self driven) 3) Collaboration and team dynamics
    - 1) work well with others 2) coach-able 3) time management/triaging & creating deadlines effectively
    - 1) Willingness to learn 2) Communication 3) Adaptability
    - 1) Willingness to learn 2) communicate/team-player 3) ability to accept failure/take criticism/incorporate feedback
    - 1) Reliability/commitment/-accountability 2) Transferable skills 3) Trainability/ Curiosity
    - Punctuality flex

2) What leadership structure and/or qualities are needed to run a successful biofoundry?
    - Multidisciplinary, specialized team underneath leadership (ie one person in automation, one for molecular biology, one coder)
    - Flat structure. Detailed technical orientation. Collaborative. Good marketing and communication. Creative funding.
    - Good communicator. Proactive. Well organized. Good fundraiser. Good recruiter.
    - general open mindedness to new and innovative ideas; giving people runway to try new things; project manager that is heavily involved, boots on ground, to keep up with what's going on; NOT MICROMANAGE; streamlined communication between leadership and people in lab and PM; flexibility in product, deliverable, process, etc
    - Clear business model, technical competence, teamwork skills, good mentorship vs management, communication with entities outside your own space, have fun
    - Science technical and administration leadership are within the team to tackle each part. The team knows where to go to each. Approachable.
    - Good communication between technical and business teams. Redundancy to mitigate single points of failure.

3) What do you believe should be considered more heavily during the hiring process: academic experience or on-the-job experience? Why?
    - on-the-job experience !! Work ethic, references more valuable
    - On-the-job experience for mid-to-senior positions, but for entry level positions, the academic experience is important.
    - Lean toward experience with automation jobs. But adequate academic training to tackle novel process challenges is essential.
    - Both are important, they need the ability to deliver projects and problem solve.
    - Depends on the needs of the company both types of experience are valuable but one type of experience cant be applied as a solution to every problem, a mix of experience in both is a healthy way to create adaptability
    - On the job. Understanding the needs and like that type of work. If academic, experience could be fine (question unclear).
    - On the job: Shows that you can GSD (get shit done); Shows that you can collaborate with other teams; Work under timelines and report out effectively

Standards

1) What is one thing that should never be standardized?
    - People
    - Human creativity
    - The human experience
    - Creativity
    - People (instinct, creativity, and emotions drive innovation)
    - Brainstorming and human thinking.
    - Anything that standardizes to the point of eliminating all ability to make mistakes in the big picture scheme — limits accidental innovation

2) What are the 3 most important automation metrics?
    - Location. Location. Location.
    - accuracy vs precision, dead volume, and time saved/etc
    - Throughput, Variation/Reproducibility, Walk away time, Quality of life improvement
    - CV, percentage decrease in hands-on time, yield in DNA concentrations
    - Scalability, Repeatability, cost vs results

- Coefficient of variance Cv, How many cups of tea can you drink in the day, Cost per sample
- Time to action. Reproducibility. Cost for budget.

3) If there is one thing that you wish was more universally standardized across biotech, what would it be?
    - Having standards!
    - Standardized buffers
    - Protocols : methods, reagents, equipment
    - Labeling, BARCODE INK
    - More standardized success metrics for given sectors (materials vs academic vs biopharma)
    - Consumables. For example, the size of wells. Results shared
    - Data capture and structure. Data is the power source and we must have standardized electrical outlets like we do in our houses to capture the data.

Sustainability

1) What do you think is the most environmentally sustainable piece of lab or automation technology that is currently on the market?
    - Grenova (tip washing) System
    - Starlabs recycling tip boxes
    - The autoclave
    - Beckman Echo: no tips, small volumes
    - Labcyte echo uses no tips
    - Vortex
    - Microfluidic that avoid liquid handling. Miniaturization

2) Do you think automation will help or hurt company sustainability in the future?
    - On demand manufacturing leads to less waste, automation in general is more efficient via limiting user error, so in general automation will help sustainability in the future
    - People will figure out how to make the new products from automation sustainable. Might even use automation to be sustainable
    - It will help by reducing unnecessary stress and in the long term help reduce time/cost (higher throughput and lower volume), as well as some health benefits, and will lead to standardization in the lab.

- Neither- automation is a tool and can exacerbate a bad strategy or improve a good strategy. You need to be thinking both about environmental sustainability and company sustainability BEFORE implementing automation.
- Do more science with the same environmental impact by not being tied to the carrying capacity of living beings and grad students with all their lunches and coffee.
- Help, environmentally can produce less waste when used properly
- Helps so the brain power is removed from repetitive processes   Both digital and physical for cycle time. Not as vulnerable to staff changes.

3) What is the easiest way a lab can become more sustainable?
    - Optimize way to reuse or eliminate plastic ware.
    - Standardizing the Reuse of pipette tips
    - Use vendors that accept recyclable waste plastic returns and biodegradable packaging
    - Reuse/recycle tips, or use software molding to reduce the amount of tests you need to do.
    - Declutter spare consumables; Turn -80's up? Controversial. Recycle tip boxes. Turn off suitable equipment when not in use. Repurposing equipment. Cold rooms needed??
    - Horizontal Integration (being able to track your process at each step -> can be automated)
    - Find ways to reduce plastic waste- tracking usage (and socializing metrics aka shame based learning), tip washing, reusing consumables, better protocol planning, recycling.

Equipment/Vendors

1) If you could add one piece of equipment to your lab for free, what would it be?
    - A unique item that no other lab near me has (ie electron microscope, sequencer) so I can charge people $$$
    - Ambr 250
    - Biological containment for automated liquid handlers.
    - On deck thermocycler
    - Fancy coffee making robot (SFO airport has one with a UR3 arm that makes lattes and dances).
    - Texan fluent. Echos. Illumina. Microscopes.
    - (Robot that makes tea)

2) What makes a vendor easy to work with?
   - Quick response, getting a full picture before throwing expensive solutions, same attitude towards all customers (high level funding vs none), open to customization, agnostic to hardware/equipment
   - Vendors should have a solid technical understanding, vendors should reply and be able to reply quickly, vendors should be open to negotiations and willing to provide detailed cost breakdown.
   - Two way communication (transparency, collaboration, really understanding needs, domain expertise in their product, responsiveness, willing to integrate with other vendors).
   - Responsiveness. Giving them money. Local support. Do "they" understand my science? Transparent pricing. Frictionless access to applications and apps support.
   - Good Communication, best swag, straightforward pricing, honesty :)
   - Accessibility. Response time. Creative invoicing issues. Being part of your team. Turnover support. Consistency/reliability. Nimble and not a lot of red tape.
   - Communication- direct line not automation and having someone who knows you. Adding value to the product you are purchasing through training/check ups/demos/contact. Updates in software / products to still work on experiments. We also like being taken out.

3) In the long-term, is it more valuable for a lab to maintain equipment in-house or pay for service contracts? Is this answer different in the short-term?
   - In the long term, it's more effective to maintain in house due to not only financial aspects but from a convenience standpoint as well, the short term however service contract may be more effective if you are using specialized equipment
   - If you are a small lab it is good to let the damp lab manage maintenance of their own facilities for your experiments in the long and short term. If you have your own machines the contract is a good bet in the long term. Maybe less so for short term usage.
   - Depends on lab’s mission (training oriented vs. CLIA requirement)
   - It depends on the response time of the service company, as well as the lab time and money constraints. It’s beneficial for labs to have service contracts in both short and long-term for turnaround time and money-saving benefits, particularly if you have to replace a very expensive hardware piece.

- Both are necessary (in both short and long term)- it depends on the particular instrument, cost of the service contract, size/specialization of the in-house automation team. It’s all about ROI and uptime.
- For a large company, in-house as it is too complicated with vendors. Trained people in the lab to repair and diagnose problems. Ownership issue in lab to fix immediately.
- Service contracts can be a disadvantage to smaller labs as too expensive. There are funds for equipment and service contracts to support labs and making equipment publicly available! Service contracts are required as we want to trust the equipment. Training a student/team member is good but they may leave resulting in more time and money to train someone new!

Automation

1) What are 3 bad or insufficient reasons to automate a workflow?

- 1) Automation for bragging rights 2) insufficient spending 3) low use case/throughput.
- 1) “I can’t get it to work manually.” 2) looks good on a resume 3) free up drinking time.
- 1) Just because you can doesn't mean you should. 2) Don’t automate a poor protocol. 3) Price per sample does not make sense.
- 1) Optics- appease investors or satisfy a grant requirement or look high-tech. 2) Getting to 100% automation for a workflow- sometimes there’s better ROI for a human to do it. 3) Automation won’t fix bad science. If it’s not working at the bench, automation won’t magically fix it.
- 1) You don't want to do it manually 2) a small number of people want it 3) something that is too variable
- 1) To be trendy / to say that you’re doing it 2) the manual procedure isn’t fully optimized itself (too early in the experimental cycle) 3) cutting corners
- 1) bragging - hype 2) spare money to be spent by x date 3) sacrificing protocol and making things worse (not thought through) with limited long term planning of integration with the lab team

2) In what measurable ways can automation increase innovation in research?

- Increase diversity of researchers who can do experiments (ie disabled persons, low-funded researchers, students)

- Time saving, reproducibility, throughput, access/training, burnout/turnover, error rates, creative time.
- Helps to find non obvious solutions, able to pursue new research using automation, throughput, repeatability
- More exploration, standardization, and digging deeper into real science. Saving time (for creativity). BIG DATA.
- Speed, walk-away time, lab throughout, better utilize personal/lab space, reduce experimental volumes, decrease contamination via hardware as well as reduce human involvement, research AI can scan data (AI do your work for you).
- Fail faster. Free up time. Increase trust in data. Democratize access. Free up time to think and communicate instead of doing busy work.
- By generating more data, decreasing variability, and freeing up time through efficiency scientists can explore a larger experimental space.

3) What is the most limiting aspect of current automation techniques/platforms?

- Being able to easily transfer knowledge between different platforms, more standards between different platforms, make it more friendly to beginners.
- Reliability. Full automation vs simple utility tasks scales up cost nonlinear. Clumsy human work like columns is hard to do by machine.
- Range of volumes (1ul and 5mL in one go), cost of dead volume, on-deck optical reads not available
- Needing a human operator to manually move things, clear use case examples, no database for troubleshooting issues, dependent on engagement with vendors
- Digital component. Staffing issues. Budgets. Socialization of the value of automation. Spatial limitations of labs.
- A lack of interoperability requiring specific domain expertise. Trade off between integration (physical) and flexibility. Systems need to be more connected in modular, scalable, reconfigurable ways.
- Being able to use large volumes and small volumes for efficiency. Constraints in materials of products matching with experimental needs. Contamination issues between wells. What about recycling consumables?

Experiments

1) What are 3 reasons why experiments fail?

- 1) Bad cells/right media 2) human errors 3) no controls
- 1) overworked 2) bad project scope 3) mislabeling samples

- 1) Expired reagents 2) Contamination 3) Liquid handling inaccuracy/imprecision.
- 1) Physical mistakes 2) communication issues 3) environmental issues
- 1) Cells die (or are killed by mistake - cellslaughter) 2) Contamination 3) Reagents are dodgy/expired
- 1) Poorly designed for example controls not implemented 2) Environmental factors 3) Fatigue.
- 1) Bad/imperfect protocol interpretation/representation 2) Insufficient replicates vs experiment variability (inconsistent results) 3) Steep learning curves

2) What are the top 2 challenges you would expect when transferring a manual protocol to an automated service lab?

- 1) Liquid classes 2) vague manual protocols.
- 1) Limitations of capabilities/bandwidth 2) I may not understand my process well enough to make others understand sufficiently
- 1) kit doesn't work with the robot 2) manual QC steps in the middle 3) non-quantifiable steps ie. determine normalization based on how bright a band is
- 1) Validation of efficiency and performance. 2) The unwritten rules/ changing human movements to robot movements with feedback to fix mistakes
- 1) The initial expectations of how the protocol is transitioned 2) having to adapt the protocol from manual to automated to account for available resources
- 1) Not detailed enough protocol to transfer. 2) Current methods for manual protocol may not be automation friendly. Need to reconfigure the protocol. Not willing to change protocol to optimize.
- 1) Expectation setting- things probably won't go well the first time and that's normal. Iteration happens at the bench and also needs to happen on automated systems. 2) Implicit and explicit steps in a protocol. Calling out the most important steps and the ones that people often mess up at the bench.

4) What types of experimental failures or challenges can automation solve?

- Contamination, variability, throughput, cost analysis, carpal tunnels, reduce redundancy (no automated human), and reduce human errors.
- De-bloating protocols with superstitious steps because they worked once (but not confirmed they caused the success of the experiment, ie correlation doesn't equal causation)
- "Did I pipette that well??" Humans make mistakes! Connecting people all over the world with the same protocols improving collaborative research. Increasing statistical power. More variables in an experiment (time course, simulation, anything you can think of!)

- Audit trail. Throughput. Fatigue. Staffing. Contamination. Reproducibility. Standardization of the process. Running unattended and 24x7.
- Human error, bigger experiments, traceability/Metadata, timing control, miniaturization cost saving makes bigger questions, 24/7 processing, multi site transfer of knowledge and techniques
- Easier to miiss non-obvious solutions, time constraints at both large and small time frames, human error, avoiding bias in data analysis
- By getting process data and metadata automatically via automation, you can start to tease apart the “unknown unknowns” of biological variability to understand experimental failure.

Software

1) Whether they currently exist or you want them to exist, name 3 software tools you need.
    - 1) Software that pings your cellphone when something goes wrong 2) Dashboard of experiment progress, totals, etc. 3) camera over your experiments to look at them remotely (like a doggy cam)
    - User friendly interface with modularity for analysis, A science Siri/Alexa, Open access graphing/stats
    - CoPilot X, the perfect/fully integrated Elab notebook, and Standards for lab work.
    - Dedicated software for comparing construct sequencing to its design specification, AI integration, tool that can tell you if you can build your design based on lab inventory
    - Closed loop experiment planning and execution (AI scientist). Magic statistics tools (tells you what the best/correct analysis methods for your experiments and data). Automation/lab system design tool. Allows you to play with the amount of automation vs humans. Creates simulations of various situations (similar to Artificial’s “digital twin” tools).
    - Software that can integrate other software. Capture and analyze. Better electronic lab notebooks. Game response.
    - HT DNA Assembly design tool. Liquid handler script remapping to translate standard processes between different platforms. Vr remote control

2) What are 3 mistakes software tools make?
    - It’s not the software that makes the mistakes but the programmers, programmers don’t talk to the biologists, and we think the question should be "what are 3 mistakes software engineers make?”

- 1) User interface (nondescript) 2) No regular feature updates/improvements/bugs 3) level of customization, templates not compatible, etc.
- 1) Poor user interface 2) too complex vs too simple 3) limited support for interoperability
- 1) it's normally the human behind the software that makes the mistake. 2) assumptions of data 3) code to user friendly language/buttons
- 1) Flaws in internal edge case logic. 2) Brittle/broken dependencies/maintenance. 3) Overfitting. (Closed source too)
- 1) User interface is not intuitive. 2) Mostly not open source. 3) Software limits the ability to analyze the raw data. Black box that does not interact with others.
- 1) Going down a rabbit hole of specialization and customization rather than more holistic feature planning. 2) Creating too many ways to perform the same operation. 3) Creating the wrong type of training content (too specific or not engaging/practical)

3) Where is the biggest deficit in the transition between software & the bench?

- Developers with a working knowledge of biology
- C++ is not English, precision in wording or getting ideas across, a translator is key (a good understanding of both fields is key), a small thing/step is crucial.
- Ambiguous manual processes. Variables not well understood. Manual processes are not always possible with machines at hand. Huma vision/proprioception/intuitions/hands dexterity not available
- Can chatgpt do it? Making the software friendly to non-code users. Curriculum changes? Inter disciplinary coming together. Open source by law?
- Going from human specifications to a language a machine can understand, documentation, communication between user and software, lack of standardization for lower level protocols, interoperability
- Communication challenges based on different jargon. Endless requirement gathering. Transition from dev to prod. Lack of connection of the lab with the software. Cost prohibitive.
- Software tools aren't developed collaboratively with a specific researcher's digital lab strategy in mind (or highlight a lack of a lab's digital lab strategy). Leads to lack of adoption and resistance to change.