Emerald Cloud Lab

Cloud Case:
Pragma Bio X Emerald Cloud Lab

I. Capabilities Expansion:
Increase efficiency via access to (normally) complex and costly instrumentation

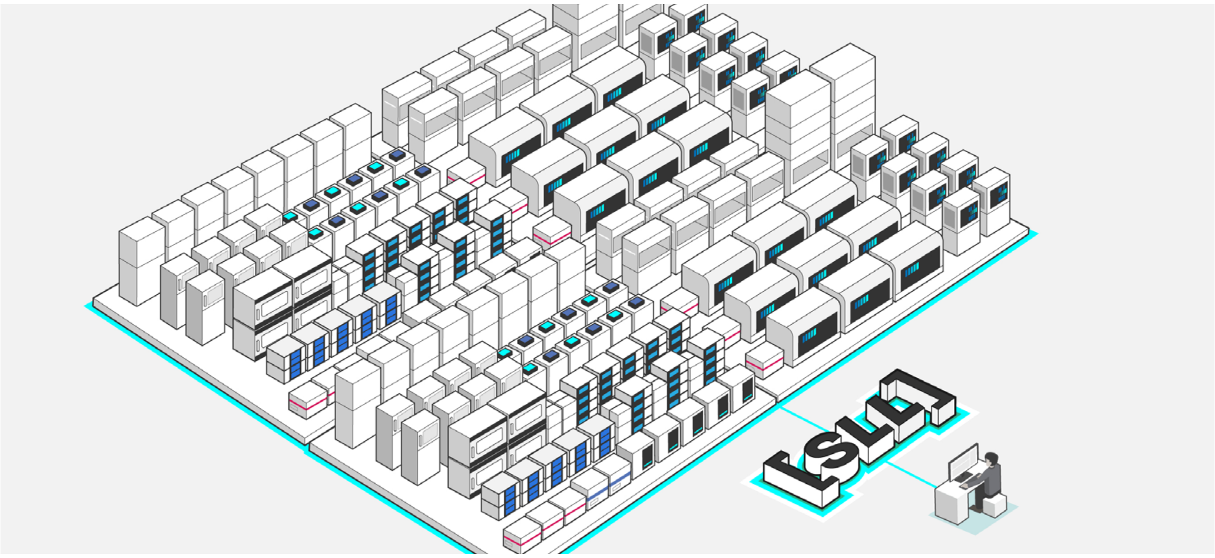
As a startup operating in the natural product space, the obvious challenge for Pragma was finding access to a diverse array of equipment that could keep up with a very wide range of potential discovery avenues. With analytical chemistry at the forefront of these needs, the requirements for instrumentation were not only broad but expensive—a truth which left Pragma with two potential pathways forward for their R&D: Cloud Lab or Contract Research Organization (CRO).

When faced with the challenge of needing diverse, high-cost instruments like LCMS (Liquid Chromatography-Mass Spectrometry), Pragma chose Emerald Cloud Lab (ECL) as a primary platform for two main reasons. First, Pragma has always been a cloud-first company and the team is very familiar with cloud engineering, DevOps, and pipeline development for bioinformatics, big data, and AI. ECL extends these same principles into laboratory work and so the paradigm is well aligned with Pragma's technical skill set. Second, Pragma's team already contains great scientists with expertise in analytical chemistry, and the intention was specifically to externalize experimental execution but to keep experimental design in-house. As such, they were looking for a solution that offered more "bare metal" access to instruments without an intermediary design service.

This illustrates a key difference between the traditional CRO model and a cloud laboratory: namely, that cloud labs allow scientists to retain complete control over all scientific decisions, and therefore expect those scientists to understand what instruments they will use for which experiments and why. This reality meshed with Pragma's goals and company style as they endeavor to foster as much in-house tacit knowledge of their experiments as possible, rather than be forced to navigate an obligatory consultative layer between protocol design and execution.

In practice, Pragma is able to use Emerald Cloud Lab's flexible Command Center software to build upon its existing "tech stack" by creating an indexable library of evergreen protocols: codified methods rendered in Symbolic Lab Language (SLL) that are wholly owned by Pragma and push-button reproducible.

Through ECL, Pragma Bio has not only accessed advanced instruments but seamlessly integrated these tools into their workflow without the financial burden of ownership. One of the first successes was developing a series of LCMS experiments, which were crucial for analyzing the complex interactions within their target molecules. The ability to quickly design, test, and iterate on these experiments using ECL's platform demonstrated a significant leap from traditional biotech methods to a more dynamic, shorter, software-like development cycle.



For example, Pragma was able to compare LCMS using triple-quadrupole and time-of-flight methods in parallel threads. Moreover, they were able to produce an optimized protocol that could detect a heterogeneous collection of natural product standards. LCMS method development is notoriously complex and can require potentially hundreds of sample injections. Through ECL, Pragma was able to run over 700 sample injections across 19 separate experiments and to apply machine learning principles such as parameter grid search to test hundreds of combinations of temperature, pressure, and voltage settings and to quantitatively determine the importance of each parameter and parameter combination on the resulting protocol.

In a traditional CRO model, this would have been exorbitantly expensive as such services are typically priced per sample. Conversely, it would have cost nearly a million dollars to procure the requisite instruments to do the same work in-house, not counting operational costs and time. Through ECL, Pragma was able to conduct this protocol optimization within one quarter and likely it would have taken longer than a quarter for in-house instruments to arrive and be set up. Importantly, they were still able to execute entirely different protocols such as NMR and Flash Chromatography during this same time period using parallel threads.

Alongside the LCMS, Pragma runs many other workflows across ECL's 240+ unique types of instruments, including:

- Western Blot
- Absorbance Spectroscopy
- MassSpectrometry (MALDI, Q-TOF, QQQ)
- Thermal Shift
- Flash Chromatography
- PCR
- Electrophoresis
- Sample Preparation
- Stock Solution
- NMR
- And reporting notebooks that run in Manifold and post data back to their AWS

II. Process Integration:
A lab that interfaces is a lab that scales

The adoption of ECL has brought more than just advanced instrumentation to Pragma Bio's now cloudsourced lab benches; it has revolutionized their entire data management and operational processes. ECL's robust Laboratory Information Management System (LIMS), fully built into Command Center, and its seamless API integrations allow for a fluid bidirectional flow of data. This system has empowered Pragma to monitor processes in real-time and access data effortlessly, which is particularly beneficial given the complex nature of their LCMS and other data and the need for continual optimization. To use process monitoring as an example, at Pragma's headquarters there is a comprehensive AWS QuickSight dashboard on continuous display throughout the office covering company-wide metrics ranging from website traffic to AI model performance to genome database size to–importantly–laboratory protocol execution, frequency, and turnaround. This dashboard is supplied both by notebooks that run within ECL Manifold and post data back to Pragma's AWS databases as well as lambda functions that periodically query data from ECL's Constellation database using the Constellation Python SDK.

The enhanced connectivity and streamlined data handling of ECL mirror the structured and review-oriented workflows of software development, including code development, review cycles, and commits. These processes have not only improved the scalability and reproducibility of experiments but have also brought a new level of precision and accountability to Pragma's research endeavors.

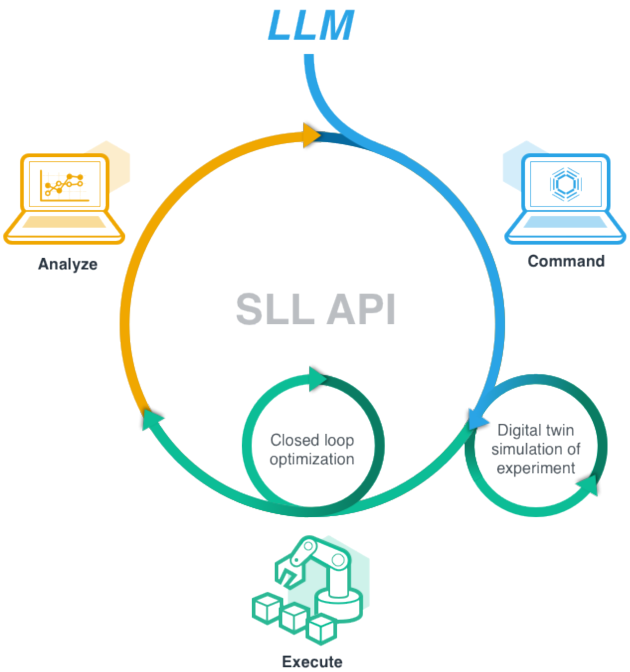
This means that Pragma can apply techniques to their science that were typically only used outside of a wet lab setting. In the same LCMS example, this looks like expansion into other technique areas such as grid search and parameter optimization of the workflows—capabilities typically reserved for performance in conjunction with machine learning. Taking this a step further, Pragma has applied these expanded capabilities to method optimization which has allowed them to a) achieve a scale of hundreds to thousands of sample injections over several days and b) swiftly aggregate and analyze the data within and between experiments—including both protocol settings and output—thanks to the codification of methods into SLL.

As exemplified by the large and complex nature of analytical chemistry experiments, in the first quarter of 2024, Pragma ran 1,208 samples across all Liquid Chromatography and Mass Spectrometry Protocols during method optimization. This ultimately required approximately 578 hours of Cloud Lab technician time over the quarter. At approximately 1 hour and 5 minutes of code development time per protocol, this same work required only 18 hours of development time from Pragma's team members. This equates to a roughly 32-fold amplification of experimental work output achieved per FTE-hour for Pragma which offers huge advantages when aiming to be efficient with a small team.

III. Leveraging What's Here & What's Next:
AI-Driven experimental design

At its root, this increase in research bandwidth via conducting experimentation at a high scale means the Pragma team can try out new protocols and instruments they otherwise wouldn't—as access to the entire scope of instrumentation comes with the subscription. In terms of research culture, this allows the Pragma team to be more scientifically aggressive and also leverage their talented software engineers to collaborate towards outcomes rather than place the burden solely on the backs of the wet lab scientists.

Stretched out over a longer arc, Pragma is excited about the potential integration of AI into its research processes. Inspired by advancements highlighted in recent publications, they are particularly interested in AI agents and co-pilots—such as the Co-scientist



agent developed by Gabe Gomes and company recently highlighted in this Nature paper—which can scale human efforts in basic science. These technologies are seen not just as tools but as partners that can assist in catalyzing many of the creative efforts undertaken by scientists already, such as drafting and reviewing protocols, tabulating finances, tracking and optimizing inventory, or even designing whole experimental campaigns.

The capability to abstract experimental designs and outcomes into programmable formats means that AI can play a pivotal role in both current and future R&D strategies. Pragma Bio envisions a near future where these AI-driven tools will not only enhance efficiency but also enable more complex, innovative explorations at a pace previously unimaginable in traditional lab settings.

IV. Wrap Up

As Pragma Bio continues to leverage ECL's capabilities, their approach to biotech research is increasingly resembling the methodologies of software development—structured, agile, and infinitely scalable. This transformation and fusion of classic biotechnology with techbio solutions like cloud labs is not just about enhancing efficiency, but is a fundamental shift towards a more integrated, technology-driven future in science.

Pragma is not just adapting to changes in the field—they are actively shaping them, for to be at the forefront of change means to also be in the thick of uncertainty. This uncertainty is, however, backed by copious amounts of data and a surge of computational power so potent, that even biotechs with 14 employees will ignite impressive change.

If you'd like to learn more about Pragma, Emerald Cloud Lab, and how you can accelerate your organization's scientific and commercial milestones with cloud labs, consider joining us on September 12 in Houston for The Cloud Lab Symposium.

If you can't wait that long, schedule a meeting with us in the meantime.

Figures

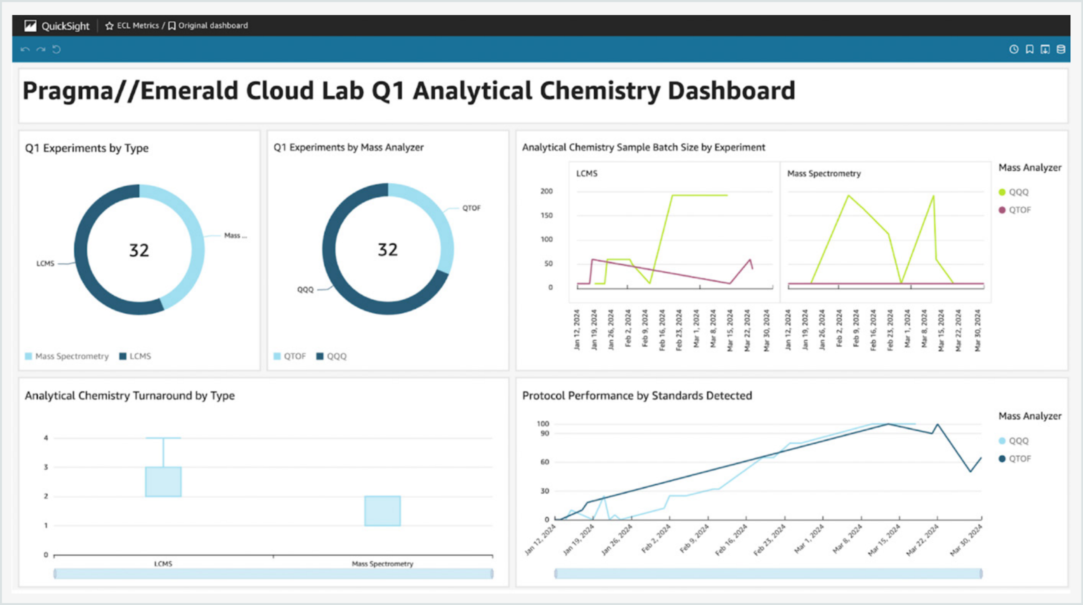


Figure 1: *Excerpt from Pragma's Dashboard to Monitor ECL Experiments*

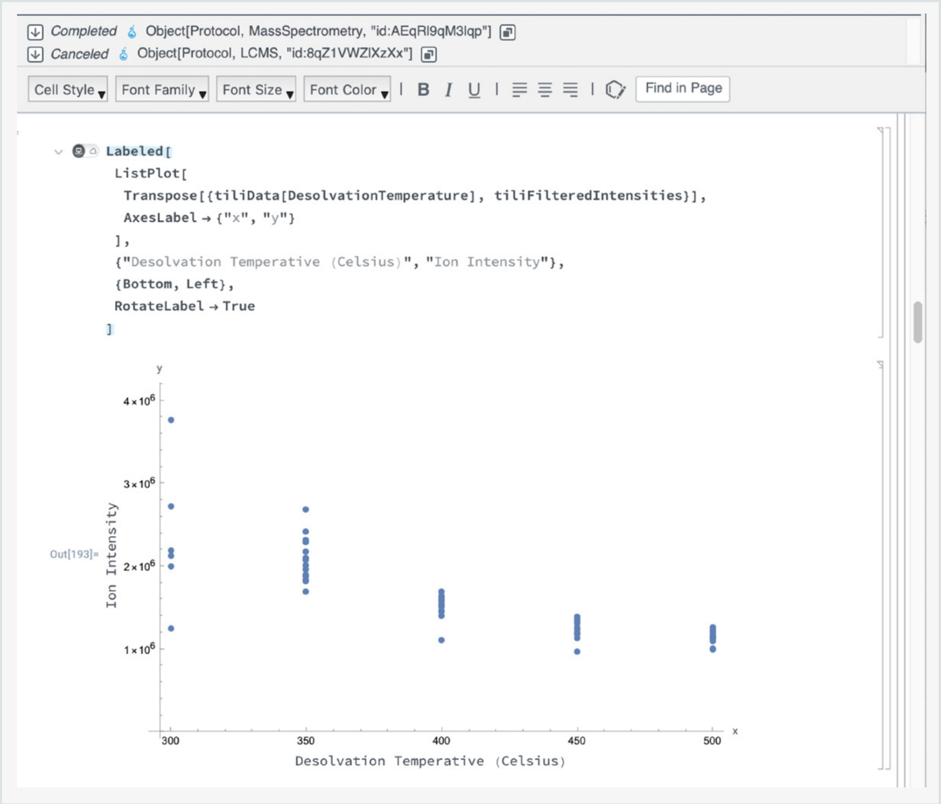


Figure 2: *Excerpt from Pragma notebook analyzing batch results for method optimization on ECL*

Evaluate the given text and produce the code snippet that accurately executes the intended process or protocol. Please consider the syntax, structure, and functionality of the ExperimentLCMS protocol. If you find any inconsistencies or errors, please point them out. The description is this:

Compose an ExperimentLCMS function to run a triple quadrupole mass spectrometer with a gradient where BufferA starts at 75% and BufferB starts at 25% and BufferA ends at 100% at 10 minutes while BufferB ends at 0% at 10 minutes. Also this should be a full scan acquisition

```
ExperimentLCMS[
    mySamples,
    MassAnalyzer → TripleQuadrupole,
    GradientA → {
        {0 * Minute, 75 * Percent},
        {10 * Minute, 100 * Percent}
    },
    GradientB → {
        {0 * Minute, 25 * Percent},
        {10 * Minute, 0 * Percent}
    },
    AcquisitionMode → MS1FullScan
]
```

Figure 3: *Example of Pragma's work in developing generative Co-Pilots for assistance in translating free-text method descriptions into executable ECL code. Below is an example of the gradient-optimized prompt being used to generate ECL-compatible code to invoke ExperimentLCMS according to a free-text method description.*

```
mySamples = {Object[Sample, "myLCMSSample91"]}
ExperimentLCMS[
 mySamples,
 MassAnalyzer → TripleQuadrupole,
 GradientA → {{0*Minute, 75*Percent}, {10*Minute, 100*Percent}},
 GradientB → {{0*Minute, 25*Percent}, {10*Minute, 0*Percent}},
 AcquisitionMode → MS1FullScan
]

Out[22]= {Object[Sample, myLCMSSample91]}

Following objects will be marked as bookmarks: {Object[Protocol, LCMS, id:8qZ1VWZlXzXx]} (Copy IDs)

Out[23]= Object[Protocol, LCMS, id:8qZ1VWZlXzXx]
```

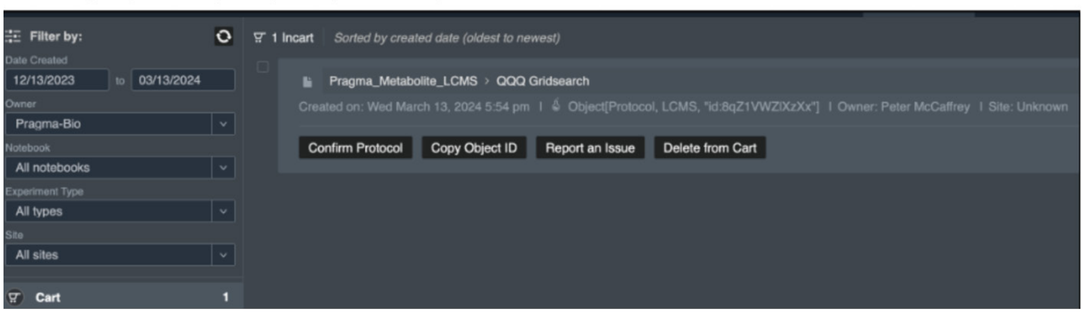


Figure 4: *Example of Pragma's work in developing generative Co-Pilots for assistance in translating free-text method descriptions into executable ECL code. The Figure below shows the code produced by the gradient-optimized prompt being successfully submitted to and run on ECL itself. The code itself is submitted using a notebook within the ECL command center (shown at the top of the image) and then the successfully compiled protocol is submitted for execution (at the bottom of the image).*