# LC-MS Data Analysis Pipelines: Scaling beyond the desktop environment

***Iman Mohtashemi*,*** *Hugo Kitano, Andrew Nichols, Seth Just, Jian Wang, Harendra Guturu, Theodore Platt, and Serafim Batzoglou*

## An automated, scalable proteomics data analysis workflow

Liquid chromatography coupled with mass spectrometry (LC-MS) has grown into a ubiquitous detection platform due to its speed, sensitivity, and applications. While instrumentation hardware continues to improve, the concurrent increase in translation from data to insight remains a bottleneck. Previously, we have demonstrated a cloud-based serverless task-based infrastructure where closed-source legacy algorithms are deployed as containerized applications leveraging AWS elastic container service. These algorithms are orchestrated with AWS services such as lambda functions and step functions. In this work, we focus on scaling label-free LC-MS data analysis workflows to enable large cohort studies using open-source algorithms leveraging distributed computing models in our AWS infrastructure.

### Challenges

- Most LC-MS data analysis solutions are built for desktop environments and are closed-source 'black-box' executables and cannot be distributed natively
- Differential proteomics data analysis of large data sets ('group runs') require data aggregation which is memory/disk limited
- Existing applications are not designed for increasing compute and memory
- There is a need to modularize the ever-growing collection of applications for both DDA and DIA acquired LC-MS data

### Solution

A carefully curated AWS proteomics data analysis workflow with choices, error handling, and exception fallbacks including:

- **Automated file transfer** to the cloud and **conversion** to standard **mzML**, **parquet** and **HDF5** filetypes
- **Automate single file analysis** for every injection upon raw data file arrival and near real-time QC
- User-specified **group run analyses** with pre-defined recipes and settings (possible with 1000s of files)
- **Spark-accelerated modular workflows** built on top of open-source platforms such as Alphapept

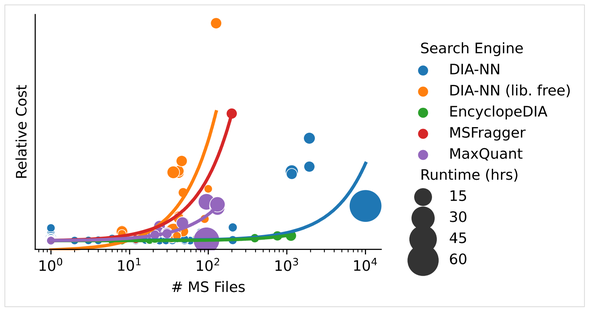
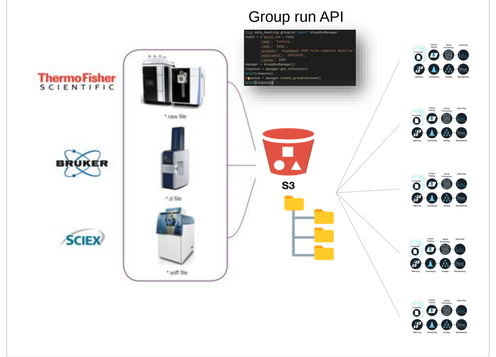



**Figure 1. Scalability and relative cost.** Our scalable search pipeline can efficiently execute **off-the-shelf search engines** engineered for Desktop execution environments. Cloud architecture enables scalability to hundreds or thousands of files by **leveraging parallel processing** and optimizing resource allocation. Total runtime is reduced significantly by employing parallelism, allowing analysts to **receive results from hundreds of files within a single eight-hour workday**. While runtime is similar across search engines, significant differences can be observed in total cost. **Library-based searches are considerably less costly and more scalable than database searches** due to the reduced search space and alignment overhead.



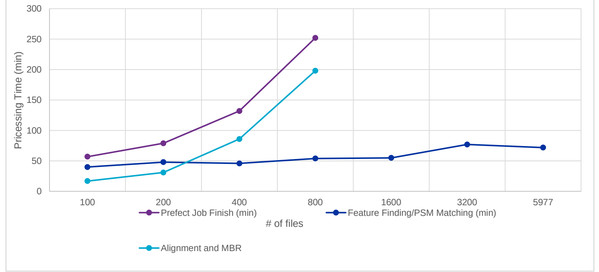
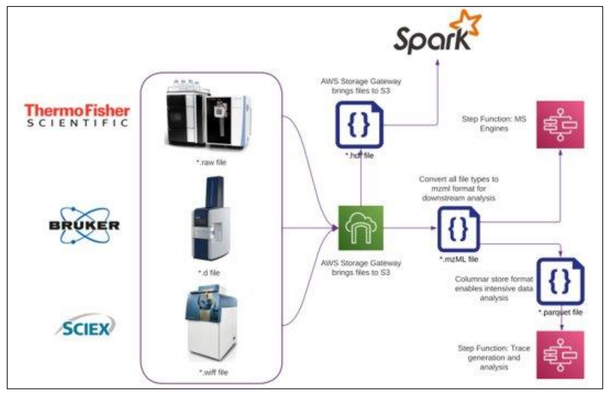




**Figure 2 Scalability.** We further experiment scalability principles. We demonstrate that individual file processing on the ECS cluster auto-scales well for feature finding and PSM matching. However, for steps that require data aggregation vertical scalability becomes a limiting factor due to size/memory constraints of a single machine requiring an alternative approach.

## Proteomics data analysis pipelines with a smart cloud infrastructure

### Multiple cloud services working in harmony

***Cloud-First Laboratory***




***Coordinating automated injection analysis***

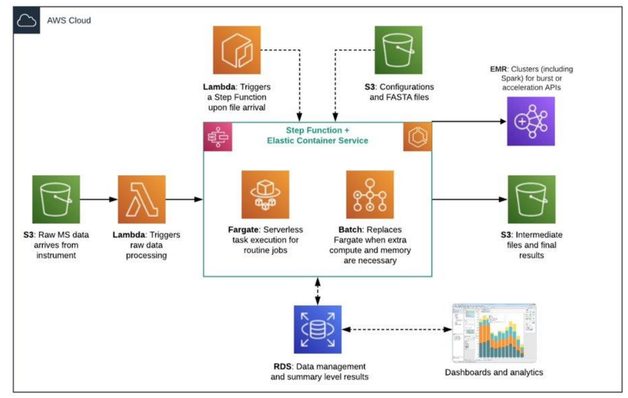



**Figure 3. Laboratory automation and cloud processing.** Vendor-neutral LC-MS files are automatically uploaded to our AWS ecosystem via transfer agents and parsed into multiple file formats for processing. Files are then processed by various algorithms deployed as nodes in the elastic container service repository and pipelines are orchestrated by the Prefect and AWS step function orchestration engines. Persistent data is stored using a combination of document and relational databases while intermediate objects (e.g., mass traces) can be stored in S3.

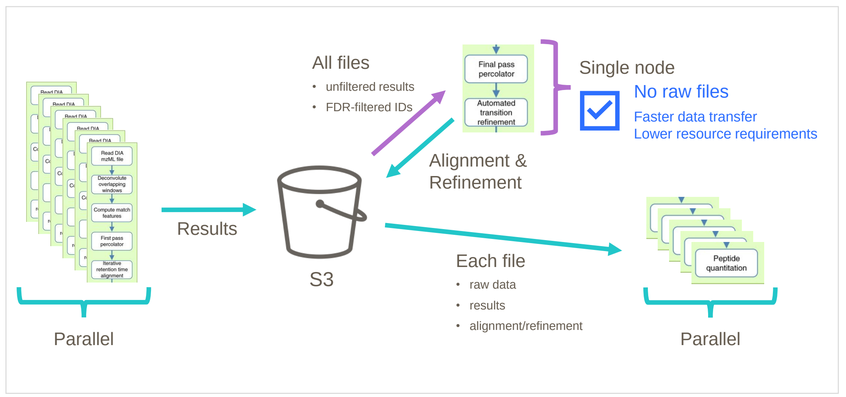



**Figure 4. Enabling parallelization of a desktop application.** **Revising** implementation or architecture of **existing algorithms** can yield significant benefits in a cloud processing environment **without changing any aspect of the algorithm or its results**. The diagram shows an approach to distribute execution of the EncyclopeDIA search, alignment, and quantification algorithm by **parallelizing loops** and **eliminating spurious data dependencies**. These changes enable searching **thousands of files** with EncyclopeDIA in **under 4 hours** with only moderate computational resources.

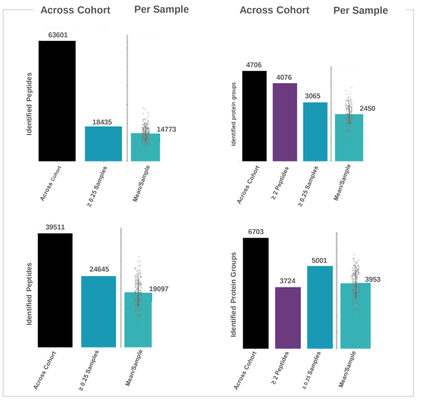



**Figure 5. Protein/Peptide identifications Example.** 200 plasma samples Alzheimer's Disease study **(Poster # PP02.129)** analyzed with 30 min DIA and 60 min DDA LC-MS analysis were processed as a single group run for each acquisition mode using DIA-NN and MSFragger respectively.

### On-demand Raw File Analysis

databricks

| Target ID | Sequence | m/z | m/z min | m/z max | RT | RT min | RT max |
|---|---|---|---|---|---|---|---|
| 1 | PEPTIDE/3 | 421.7578 | 421.7536 | 421.762 | 11 | 9 | 13 |
| 2 | PEPTIDE/2 | 629.3477 | 629.3414 | 629.354 | 11 | 9 | 13 |

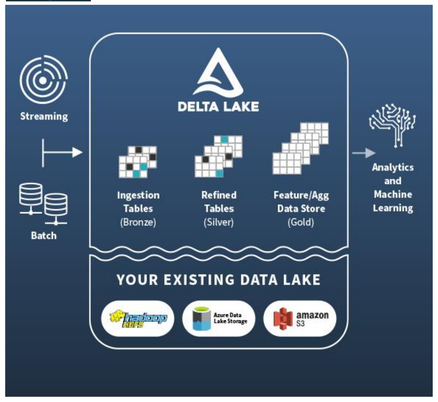



**Figure 6. Peptide-centric search.** We further experimented with the Delta Lake architecture. Briefly, raw MS files are automatically converted to the parquet format and directly loaded to delta tables. Raw chromatogram data is then interrogated using transformation functions by applying the following steps:

1. Create spark dataframe from list of target peptide m/z
2. Define a set of data transformation functions to set XIC boundaries
3. Transform the XIC targets
4. Read in the MS data, keeping track of input file name
5. Define MS data transformation functions
6. Transform the MS data
7. Define functions to facilitate join
8. Apply a range join

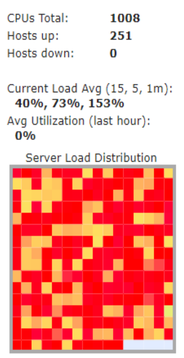



**Figure 7. Apache spark with Databricks.** Target value extraction using elastic computing. We demonstrate XIC of multiple charge states of a peptides across 7700 raw MS files in under 5 minutes. Databricks autoscaling function allows clusters to scale to arbitrary size depending on the workload.

### Acknowledgements

*Tejal Choudhary[1], Maxamillian Strauss[2], and Vadim Patsalo[3]*
*[1] NexgenInvent*
*[2] University of Copenhagen*
*[3] Databricks*

### Conclusion

- Handle hundreds to thousands of samples
- Analyze >5000 injections
- > 5000 protein groups
- Fast raw signal interrogation

- The 200-sample DDA and DIA study was processed, generating unbiased proteomics data of over 5,000 proteins
- Containerization and tranching of files per container resulted in excellent scalability of feature finding and peptide spectral matching of the Alphapept pipeline across the AWS ECS task infrastructure
- Modifying DIA pipelines to efficiently leverage cloud resources was implemented
- Ingestion of raw MS data into the Delta Lake provides a basis for ultra-fast raw signal interrogation with elastic computing features such as autoscaling

### References

*[1] Blume et al. Nat. Comm (2020)*
*[2] Strauss et al. BioRxiv (2021)*



[1]Seer, Inc., Redwood City, CA 94065, USA | *imohtashemi@seer.bio