



Cell Culture Offline Sample Monitoring

Brad Keigwin

Overview



CHALLENGE

- Disparate non-standardized data across sites
- Difficulty monitoring upstream KPI's in a timely manner
- Time consuming manual data entry
 - Prone to delays and human error



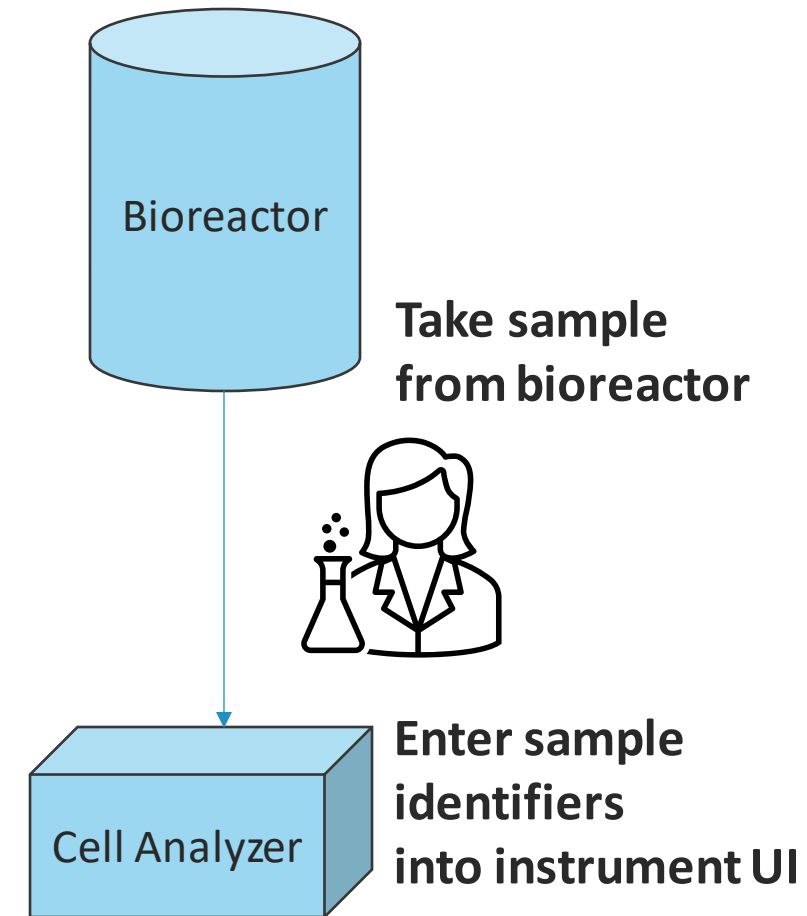
SOLUTION

- Automate the collection of data, removing the manual process and removing data entry delays
- Harmonize and standardize multi-site multi-product data
- Automatically add batch context from ERP system to enrich the data



RESULTS

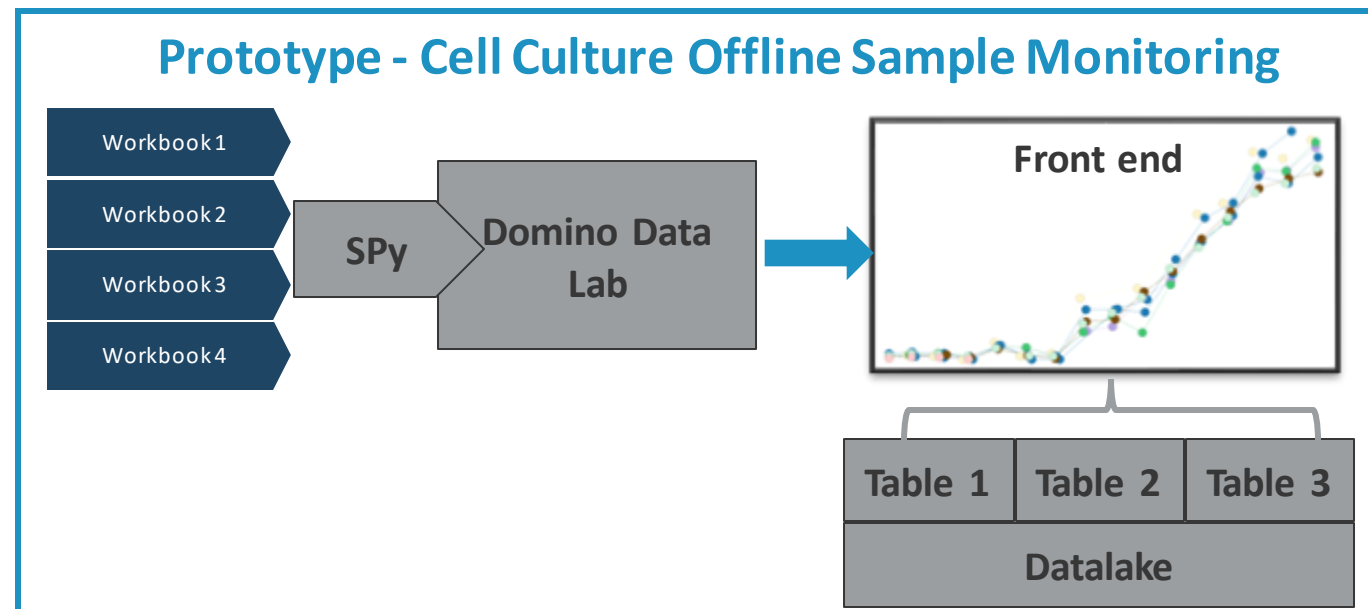
- Standardized monitoring of 17 products and 65 bioreactors across the network
- Cleaned data set living in the cloud
 - For consumption for other use cases, ex: MVA/PLS
- Saved FTE hours



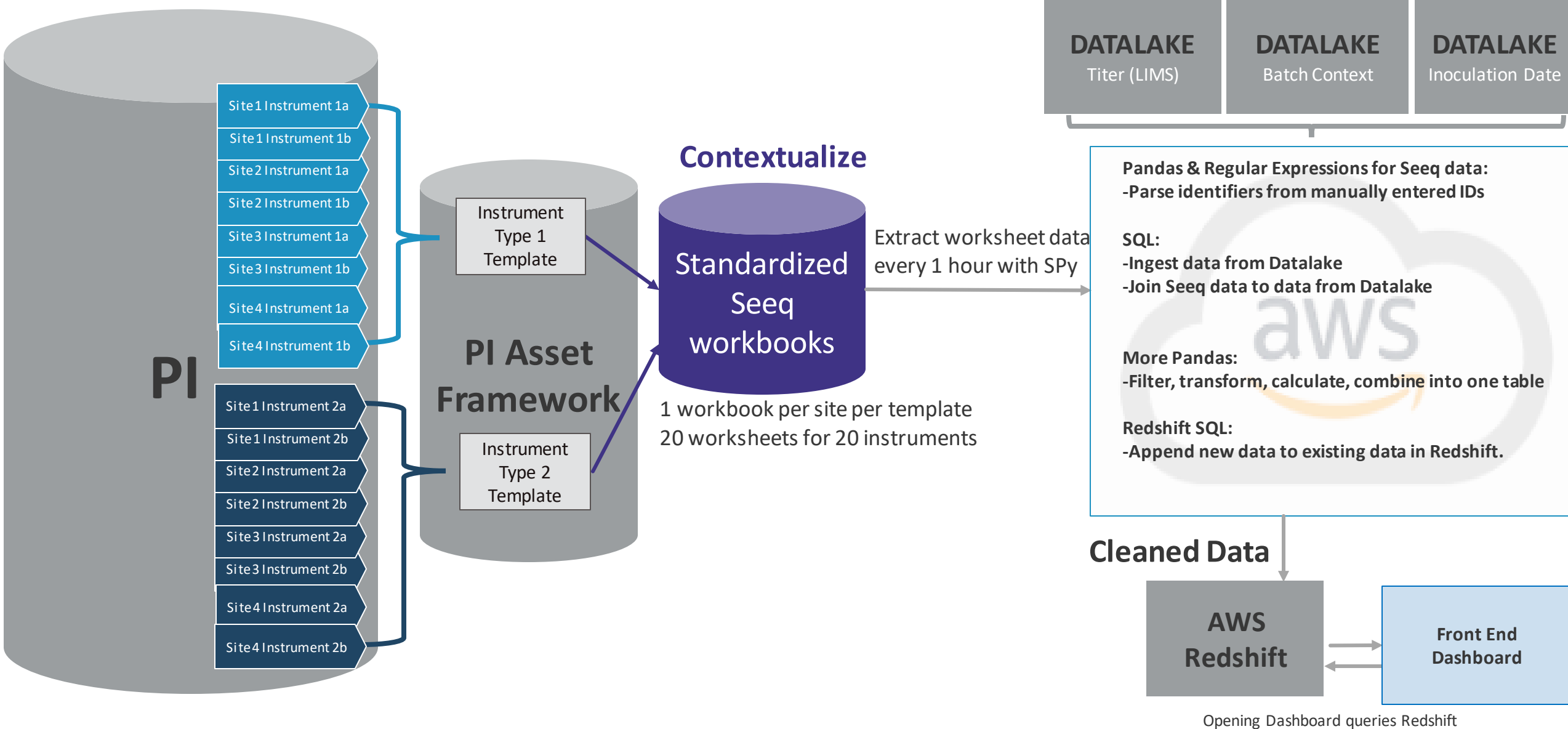
From Prototype to Scalable Solution

- I've worked on many other projects querying our data lake.
 - I know how to query our ERP system
 - Batch context, material numbers
 - I know how to retrieve PI AF event frame timestamps.
 - Inoculation date per bioreactor
- I'm familiar with dashboarding software.
 - I was able to quickly create front-end visualizations and UI.
 - Design the table structure in a way that will make visualization easy.

- This is my third pipeline reading Seeq data using SPy.
 - The first project was basically Cell Culture Offline Sample Monitoring for one site, for 6-7 products.
 - The second project was our first pipeline reading Seeq data using AWS Glue.



New Solution Architecture



Building the workbooks

Each instrument at each site gives us up to 3 identifiers per sample.

- Instrument Type 1 has 3 separate ID tags. Instrument Type 2 has 1 ID tag containing up to 3 different IDs.
- Some sites might only give us 2 out of 3 ID's.

We can build a condition intersecting the 3 IDs.

- `ID1.toCondition().intersect(ID2.toCondition()).intersect(ID3.toCondition()).removeShorterThan(1min)`

Each instrument measures a set of variables per sample.

- One sample measures N variables.
- Continuous signal data → discrete capsule table (1 row per sample)
- `Variable1.toDiscrete() ... VariableN.toDiscrete()`

After building 1 worksheet for 1 instrument at 1 site, duplicate and use Asset Swapping!

- Each site has a different formula for the condition.

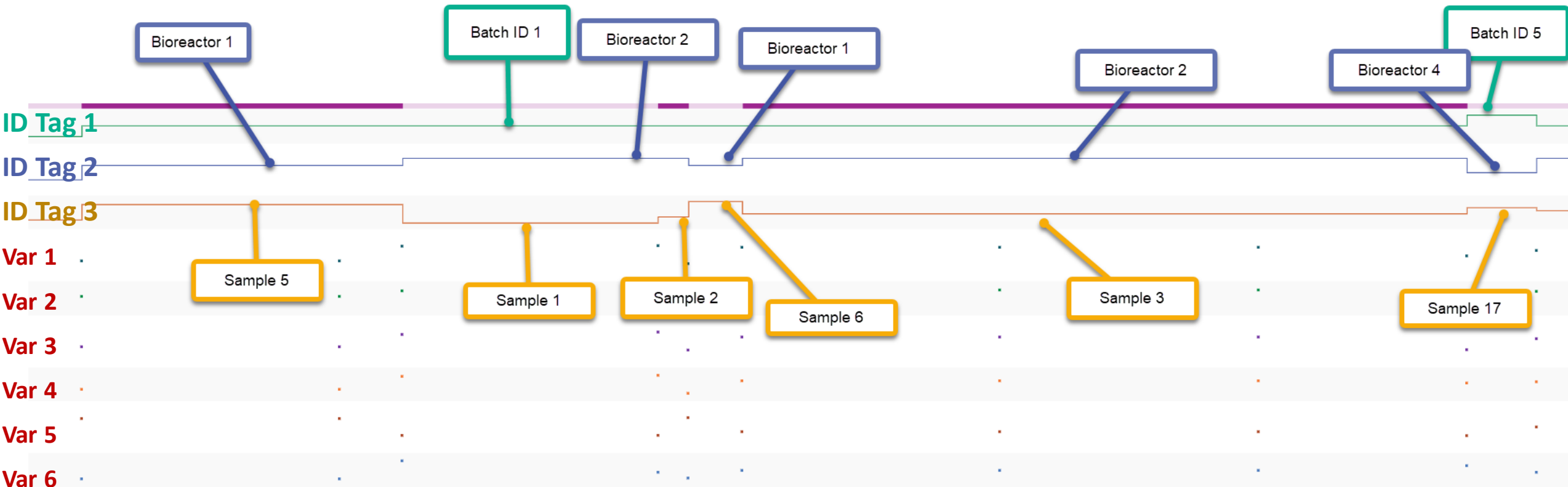
Pull the capsule data using SPy.

- Since the worksheets are nearly identical, this simplifies the code.

Capsule Creation Logic

ID1.toCondition().intersect(**ID2**.toCondition()).intersect(**ID3**.toCondition()).removeShorterThan(1min)

Take Last value per variable within capsule



[illegible]

Parsing Identifiers with Regular Expressions / Python

Unstructured Data

(manually entered text IDs into Instrument UI)



Cleansed Standardized Format

Site	Instrument Type	Tag ID 1	Tag ID 2	Tag ID 3	BMS_ID_1 (Parsed)	BMS_ID_2 (Parsed)	BMS_ID_3 (Parsed)	BMS_ID_4 (Parsed)
Site 1	Type 1	F_250L_X6	1002345	BR11	1002345		BR11	250L
Site 1	Type 2	1002345_F_250L_X6	n/a	n/a	1002345			250L
Site 2	Type 1	Sample 1/3 Day 4	AB123	Bioreactor55		AB123		
Site 2	Type 2	Bioreactor55 10Hours AB123	n/a	n/a		AB123	Bioreactor55	
Site 3	Type 1	AZ789-10K-D3S1	Not used/not configured	BR77		AZ789	BR77	10K
Site 3	Type 2	AZ789-10K-D3S1	n/a	n/a		AZ789		10K
Site 4	Type 1	1005678-0110- Bioreactor99-5	Not used/not configured	Not used/not configured	1005678		Bioreactor99	
Site 4	Type 2	1005678-0110- Bioreactor99-5	n/a	n/a	1005678		Bioreactor99	

Fill in the blank BMS_ID's by joining to tables in the datalake, or transforming with Python/Pandas.

(all ids are dummy ids)

Parsing Identifiers with Regular Expressions / Python

Unstructured Data

(manually entered text IDs into Instrument UI)



Cleansed Standardized Format

Site	Instrument Type	Tag ID 1	Tag ID 2	Tag ID 3	BMS_ID_1 (Parsed)	BMS_ID_2 (Parsed)	BMS_ID_3 (Parsed)	BMS_ID_4 (Parsed)
Site 1	Type 1	F_250L_X6	1002345	BR11	1002345	AY932	BR11	250L
Site 1	Type 2	1002345_F_250L_X6	n/a	n/a	1002345	AY932	BR11	250L
Site 2	Type 1	Sample 1/3 Day 4	AB123	Bioreactor55	1008390	AB123	Bioreactor55	1500L
Site 2	Type 2	Bioreactor55 10Hours AB123	n/a	n/a	1008390	AB123	Bioreactor55	1500L
Site 3	Type 1	AZ789-10K-D3S1	Not used/not configured	BR77	1006395	AZ789	BR77	10K
Site 3	Type 2	AZ789-10K-D3S1	n/a	n/a	1006395	AZ789	BR77	10K
Site 4	Type 1	1005678-0110- Bioreactor99-5	Not used/not configured	Not used/not configured	1005678	AG381	Bioreactor99	40L
Site 4	Type 2	1005678-0110- Bioreactor99-5	n/a	n/a	1005678	AG381	Bioreactor99	40L

Fill in the blank BMS_ID's by joining to tables in the datalake, or transforming with Python/Pandas.

(all ids are dummy ids)

Mapping Tables

- Mapping tables allow us to pull specific info from our datalake
 - Extract only what we need
 - Then combine and enrich the Seeq data
- Add more codes to mapping tables to look up more products.
 - Underlying code and data structure will remain the same
- Mapping tables are documented knowledge
 - Can be referenced / reused for other use cases

Site	Product Name	Material Code
1	Product A	Material1579
2	Product B	Material9237
3	Product C	Material6502
4	Product D	Material0132

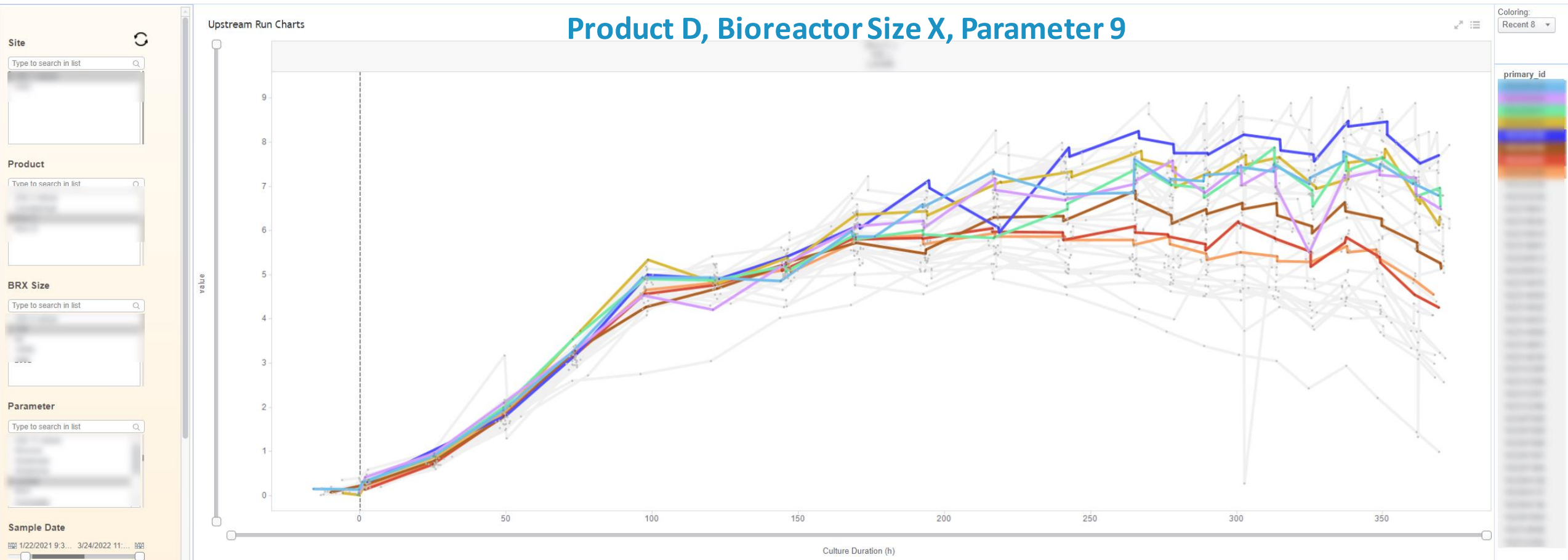
Site	Bioreactor ID	Operation Name	Operation Code
1	BR99	Inoculation	INOC_123
2	BR88	Inoc	GROWTH_STAGE1
3	Bioreactor1	7915793	INOC:1:56
4	Bioreactor55	CELL_CULTURE	CELL_GROWTH-3

Site	Product Name	Test Name	Test Code
1	Product A	TITER	Titer_Final
2	Product B	Titer (2)	Titer_2_DP
3	Product C	9837496	TITER_3
4	Product D	Titer	TITER_DAY9

(all ids are dummy ids)

Front end example

Culture duration on the X axis, Parameter value on the Y axis. Each data point is 1 sample
One line per batch. Recent batches in color and historical batches in grey for comparison
One-size-fits-all visualization



Conclusion

- **How was Seeq used**

- Data exploration. Helping to identify inconsistencies per site per instrument.
- Create standardized worksheets with asset swapping
- Easily transform continuous signal data into a nice discrete capsule table
- Read capsule table using SPy

- **Benefits**

- Generic read Seeq worksheet into AWS function for use in other projects
- Standardized upstream KPI process monitoring for all products across 4 sites
- FTE hours saved
- Cleaned data source in the cloud

- **Next steps**

- Going live in mid May
- Adding more molecules to the pipeline
- Interested to see how users will change from manual data entry.
- Adding more parameters in a future release (amount of Feed added)



For more information and event updates,
please visit seeq.com