What is Lineage?
Example: Sample Derivation

- Represents creation of daughter samples (e.g. aliquoting)
- Establishes hierarchy of data and samples along with metadata
- Linked to their parents via an experiment run
  - *(not shown on the graph)*
Example: Sample Derivation

ES-2

Overview Lineage

Expression Systems
Project Apollo - M-1 Target

Host Cell Line  HEK293
Constructions  C-1
Intended Molecules  M-1

Expression System Samples Children (2)
- ES-2.1
- ES-3-20160203-4
Example: Sample Derivation

- Splitting
- Pooling
- Combined nodes for >5 siblings
Example: Assay + Transform (Summary)

Models assay process:
samples go in machine, file spit out, transformed file out
Example: Assay + Transform (Details)

Internal structure of run captures the transform step

Final outputs have bold borders
Lineage is Experiment Runs

- Lineage backed by Experiment Run graph

- Experiment Run composition
  - Protocol Type - “Sample Derivation” or “MyAssay”
  - Inputs - Data or samples
  - Steps
  - Outputs - Data or samples
Experiment Run Use Cases

- Sample derivation
  - Splitting, pooling samples
  - Create parent/child lineage sample and data
- Assay Runs
- Pipeline Jobs sometimes
  - ETLs, Script pipelines
- XAR XML
  - Import/export Assay/SampleSet definitions, Runs
Creating Sample Lineage
Creating Sample Lineage

- Derive samples in LabKey Server UI
- Derive samples in Sample Manager UI
- Lineage columns for import/insert
  - Bulk import tsv
  - LABKEY.Query.insertRows
- LABKEY.Experiment.saveBatches/saveRuns
  - saveRuns - new in 19.3!
Deriving Samples: Sample Manager
Deriving Samples: Bulk Import

- Special columns for import
  - MaterialInputs/<SampleSetName>
  - DataInputs/<DataClassName>

- Comma separated list of names

<table>
<thead>
<tr>
<th>Name</th>
<th>SampleDate</th>
<th>MaterialInputs/Samples</th>
<th>MaterialInputs/MixtureBatches</th>
<th>DataInputs/CellLine</th>
</tr>
</thead>
<tbody>
<tr>
<td>SampleA</td>
<td>2019-09-10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CellExample1</td>
<td>2019-09-10</td>
<td></td>
<td></td>
<td>CHO-K1</td>
</tr>
<tr>
<td>CellExample2</td>
<td>2019-09-10</td>
<td>CellExample1,SampleA</td>
<td></td>
<td>B-20190308-145</td>
</tr>
</tbody>
</table>
Deriving Samples: Bulk Import

**Overview**

**Lineage**

- **B-20190308-145**
  - **Mixture Batches**
  - **Expired**
  - **Consumed**
  - **Recipe Amount** 8.0 L
  - **Recipe Actual Amount**
  - **Recipe Amount Unit** L

**Ingredients Parents**
- I-103
- I-58
- I-50

**Mixtures Parents**
- TBS

**Samples Children**
- CellExample2
LABKEY.Query.insertRows({
    schemaName: "samples",
    queryName: "Samples",
    rows: [{
        name: "Example1",
        sampleDate: "2019-09-10",
    },{ 
        name: "Example2",
        sampleDate: "2019-09-10",
        "MaterialInputs/Samples": "Example1,CellExample1"
    }]
});
LABKEY.Experiment.saveRuns({
    protocolName: LABKEY.Experiment.SAMPLE_DERIVATION_PROTOCOL,
    runs: [{
        name: "derive example",
        materialInputs: [{
            // or get existing sample by row id
            id: 105
        }],
        // get existing sample by name
        name: "ES-1.205",
        sampleSet: {
            name: "ExpressionSystemSamples"
        }
    }],
    materialOutputs: [{
        // create a new sample
        name: "Example3",
        sampleSet: {
            name: "Samples"
        },
        properties: {
            SampleDate: "2019-09-10"
        }
    }]
});
Accessing Lineage
Accessing Lineage

- LabKey Server run graph UI
- LabKey Biologics / Sample Manager lineage UI
- LABKEY.Experiment.lineage() API
- Using Query to access lineage
- Using Lineage Query filter
Example: Lineage API

- Fetch lineage from a starting seed
- Indicate direction and depth
- Returns a graph of nodes

```javascript
LABKEY.Experiment.lineage({
  lsid: 'urn:lsid:labkey.com:Sample.31.Samples:KK01_t19119',
  parents: true,
  children: false,
  depth: 3
});
```
Example: Lineage API Response

```json
{...
    "urn:lsid:labkey.com:Sample.31.Samples:KK01_t19119":{
        "children": [],
        "cpasType": "urn:lsid:labkey.com:SampleSet.Folder-3235:Samples",
        "name": "KK01_t19119",
        "parents": [ "urn:lsid:labkey.com:Run.Folder-3235:6031ac4f" ],
        "rowId": 1092270,
        "type": "Sample"
    },
    "urn:lsid:labkey.com:Run.Folder-3235:6031ac4f":{
        "children": [ "urn:lsid:labkey.com:Sample.31.Samples:KK01_t19119" ],
        "cpasType": "urn:lsid:labkey.org:Protocol:SampleDerivationProtocol",
        "name": "Derive 3 samples from KK01_t19114",
        "parents": [ "urn:lsid:labkey.com:Sample.31.Samples:KK01_t19114" ],
        "rowId": 292262,
        "type": "Run"
    },
    "urn:lsid:labkey.com:Sample.31.Samples:KK01_t19114":{
        "children": [ "urn:lsid:labkey.com:Run.Folder-3235:6031ac4f" ],
        "cpasType": "urn:lsid:labkey.com:SampleSet.Folder-3235:Samples",
        "name": "KK01_t19114",
        "parents": [],
        "rowId": 1092265,
        "type": "Sample"
    }
}```
Lineage Query Columns

- Hidden Inputs and Outputs FK to lineage tables
  - Available on Run, Material, Data tables
  - Multi-valued foreign keys

- Performance implications
  - Lots of effort spent on lineage column perf
  - Lineage executed at render time
    - Sort/filter not possible
Example: Lineage Columns
Lineage Query Filters

- LABKEY.Filter.Types.EXP_CHILD_OF
- Attach filter to an LSID column
- Filter value is LSID of a parent sample or data
  - query.LSID~exp:childof=<parent-LSID>
demo time
Resources

- LabKey Documentation
  - Experiment Terminology
  - Sample Derivation
  - Sample Status Demo - ⭐ Premium Resource

- JS API docs
  - LABKEY.Query.importData - bulk import tsv
  - LABKEY.Experiment.saveBatches
  - LABKEY.Experiment.lineage
Extra Content
Example: XarTutorial Ex. 6 (Summary)

Run summary with inputs and outputs

Example 6 (MS2 with pooling, fractionation)

MzXML file (0)  MzXML file (1)  MzXML file (2)  MzXML file (3)  MzXML file (4)
Lots of internal steps with intermediate samples and files
Example: Script Pipeline Run + Assay

RNASeqMatrixDemo
Example: Script Pipeline Run