Alexander Lex
@alexander_lex
http://alexander-lex.net

Interactive Visualization in Biological Data Science: From Bespoke Tools to Reusable Libraries

visualization design lab

TU Graz
Graz University of Technology

visualization
design lab

http://visdesignlab.net

TU Graz
Graz University of Technology

THE UNIVERSITY of UTAH

datavisyn

Based in Linz

Data visualization solutions for pharmaceutical industry

35 people and growing!

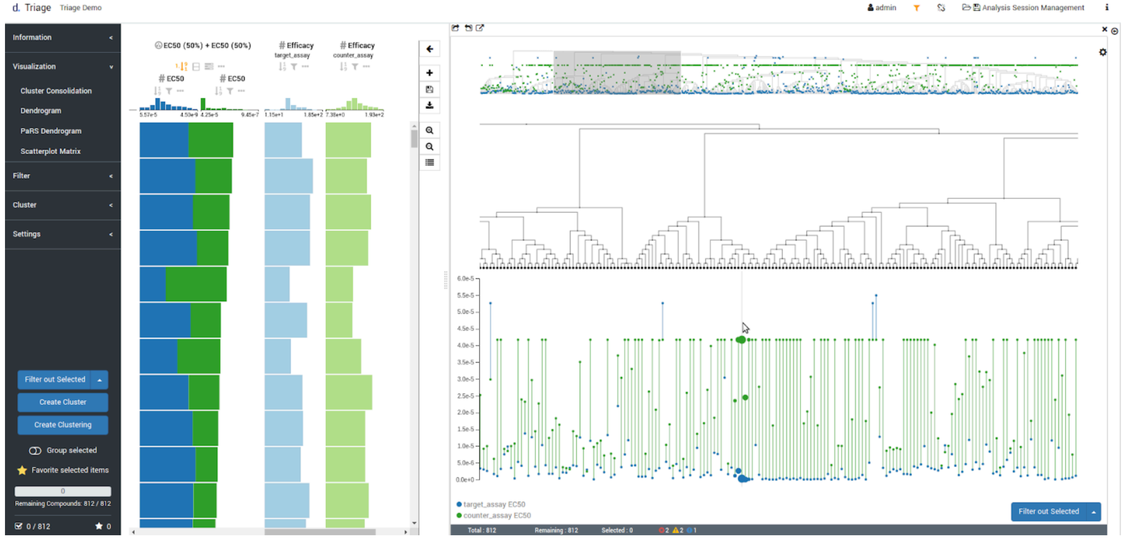
Dominic Giradi, CEO

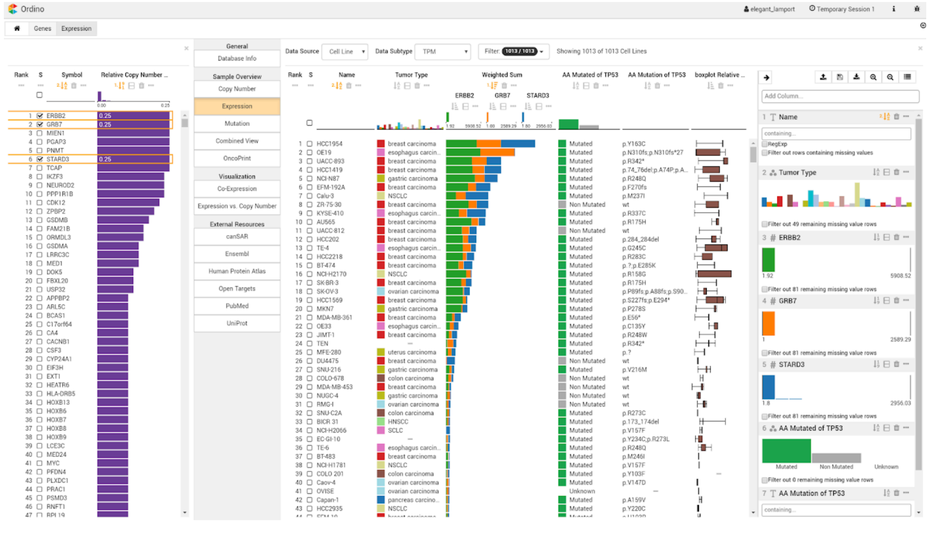
Marc Streit, CEO

Alexander Lex

Nils Gehlenborg

Samuel Gratzl





Visualization =

Human Data Interaction

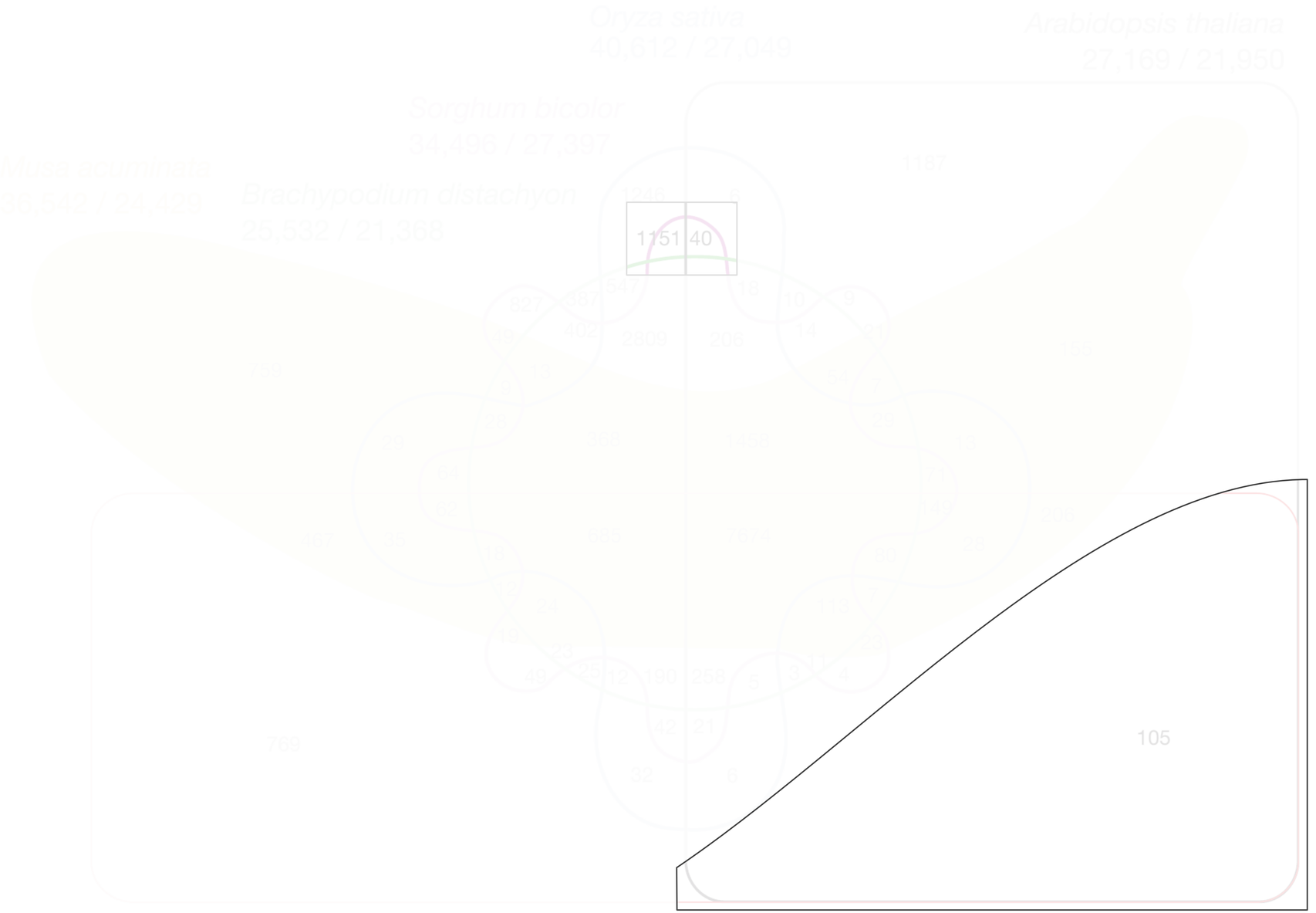
visualization

The purpose of computing is insight,
not numbers.

pictures

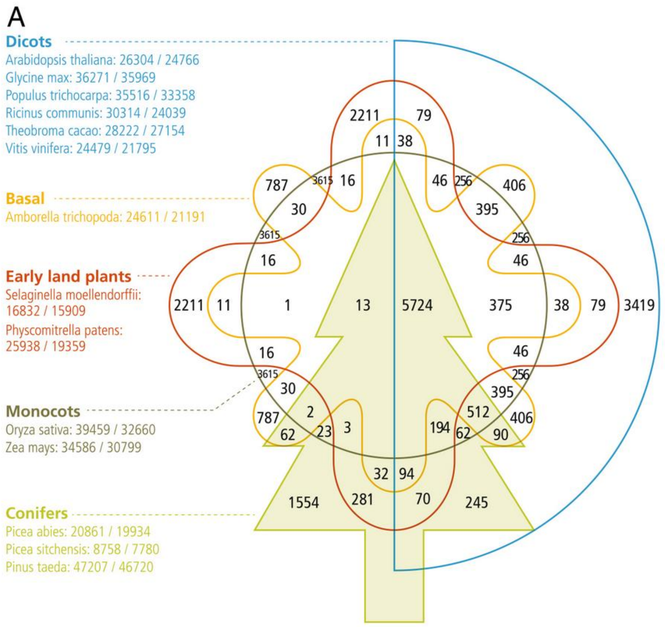
[Card, Mackinlay, Shneiderman]
[Richard Wesley Hamming]

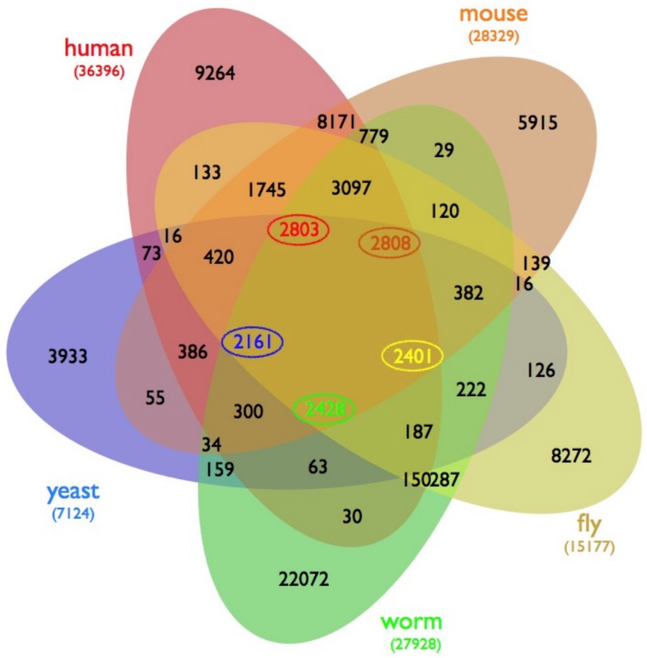
Banana	*M. acuminata*
Date	*P. dactylifera*
Cress	*Arabidopsis thaliana*
Rice	*Oryza sativa*
Sorghum	*Sorghum bicolor*
Brome	*Brachypodium distachyon*



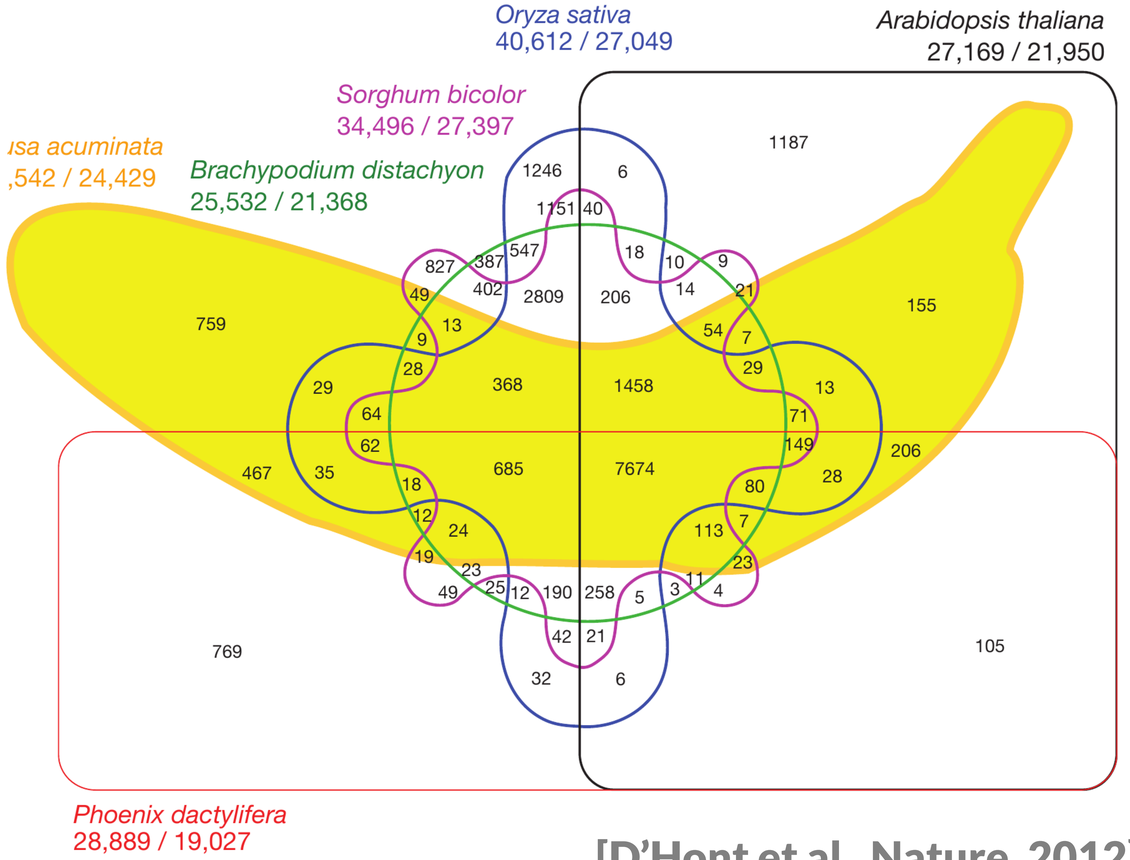
[D'Hont et al., Nature, 2012]

[Neale et al., BMC Genome Biology, 2014]

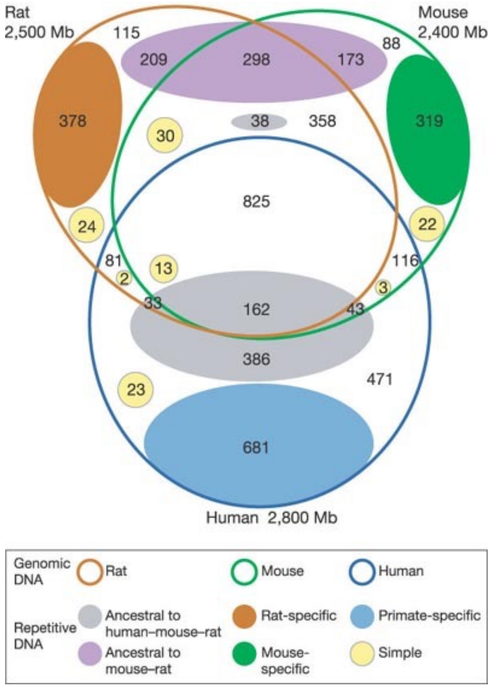




[Wiles et al., BMC Systems Biology]

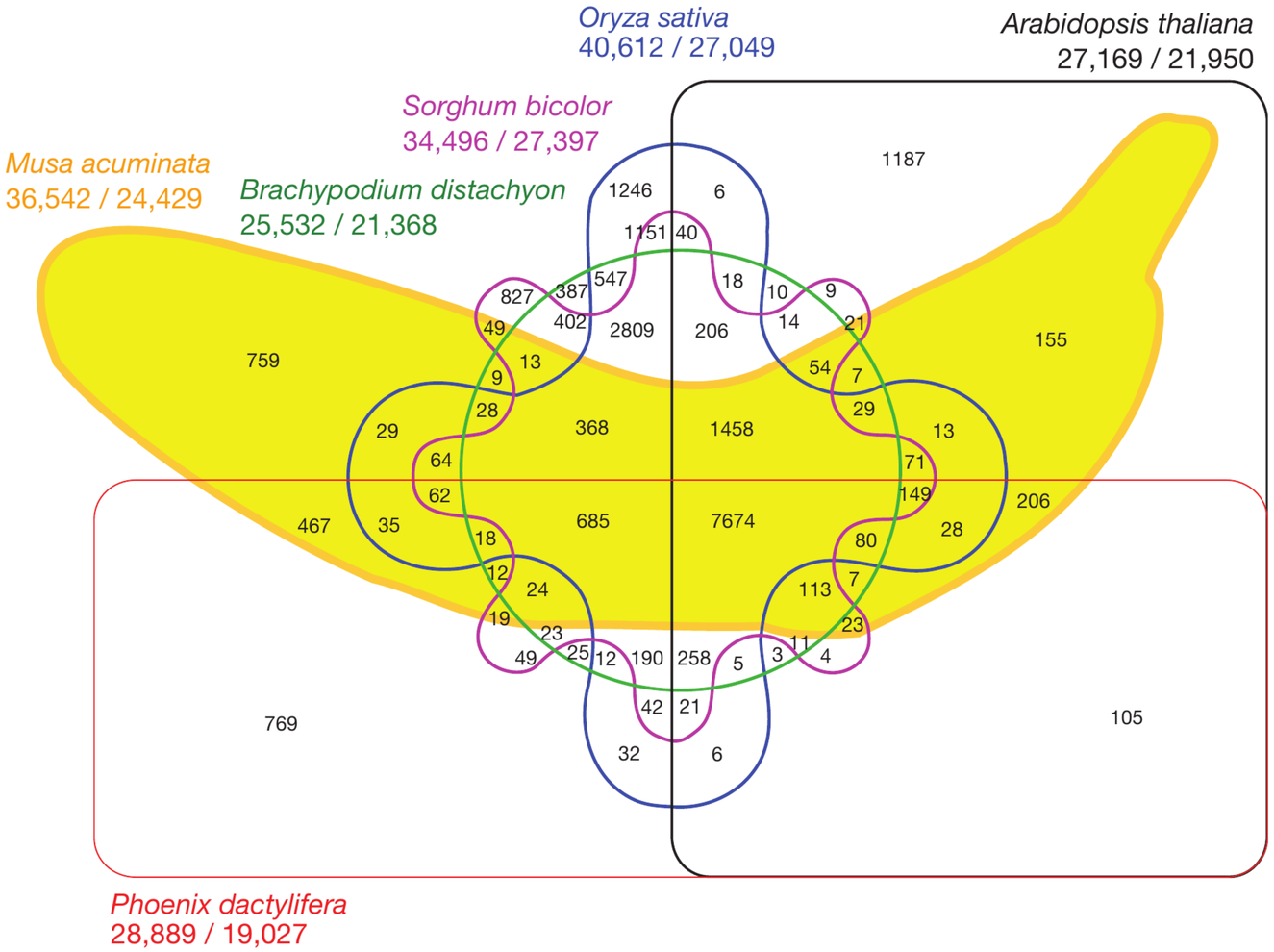


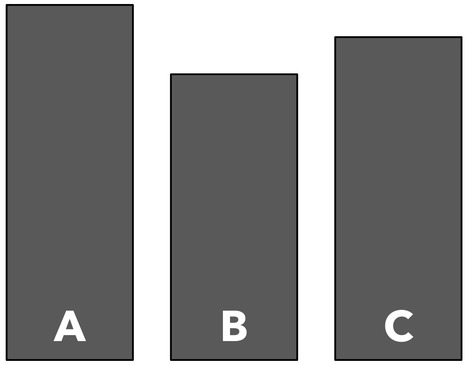
[D'Hont et al., Nature, 2012]

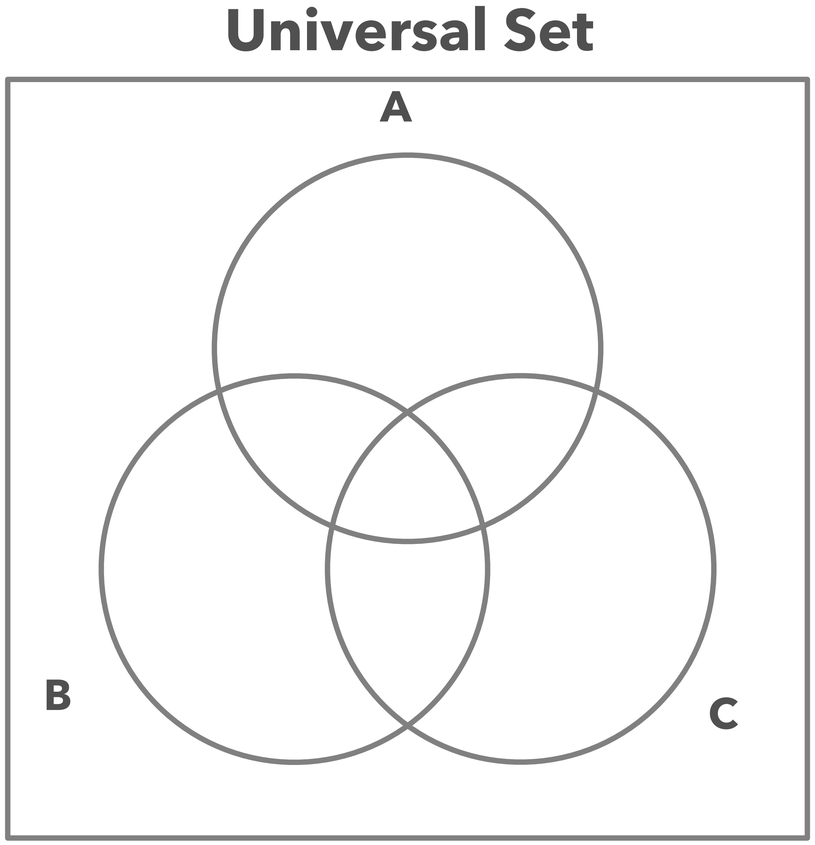


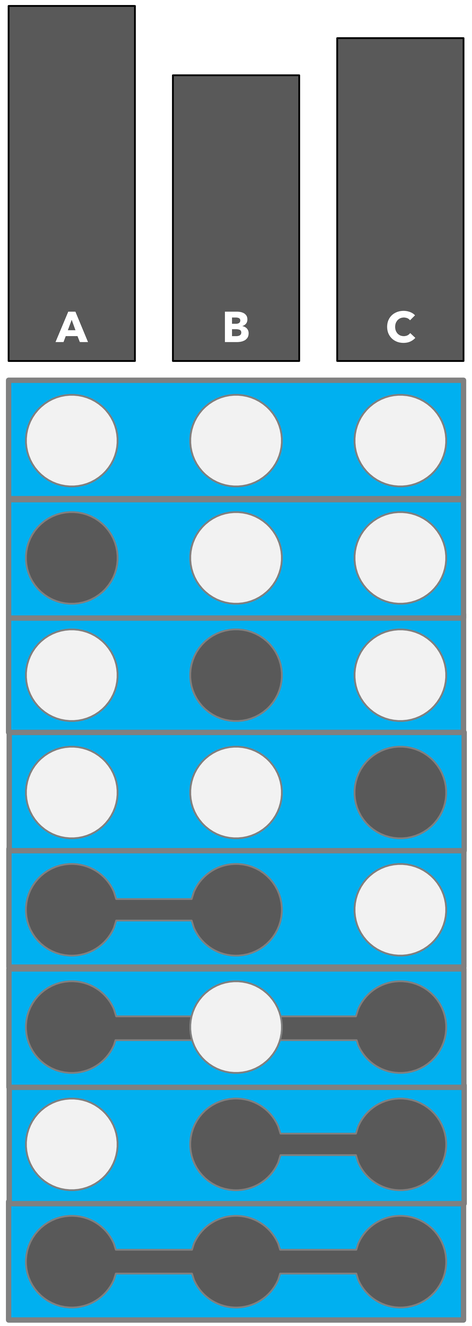
[Gibbs et al., Nature, 2004]

SO CAN WE DO BETTER?

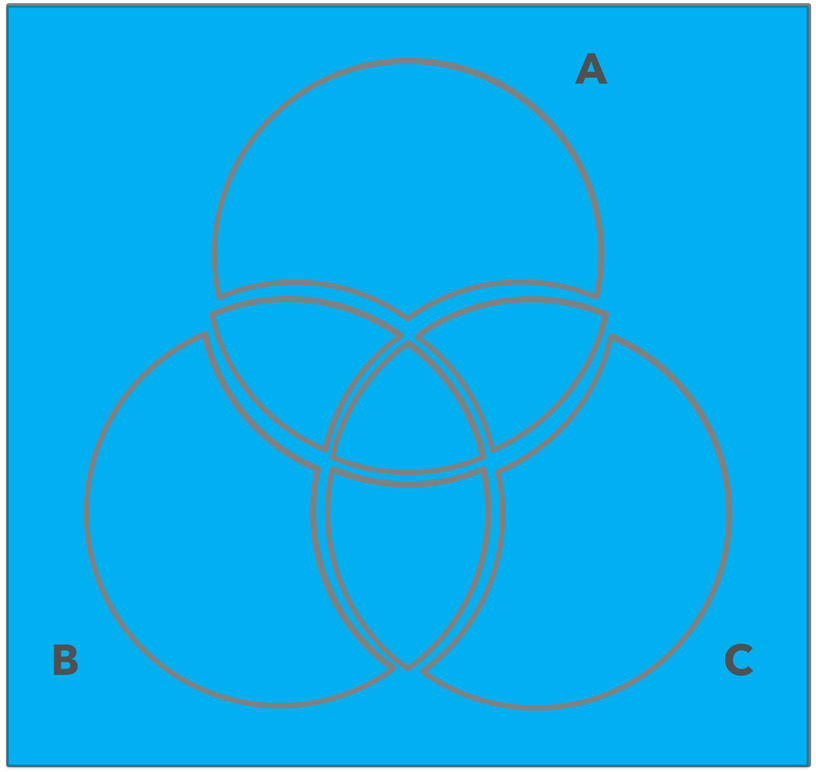


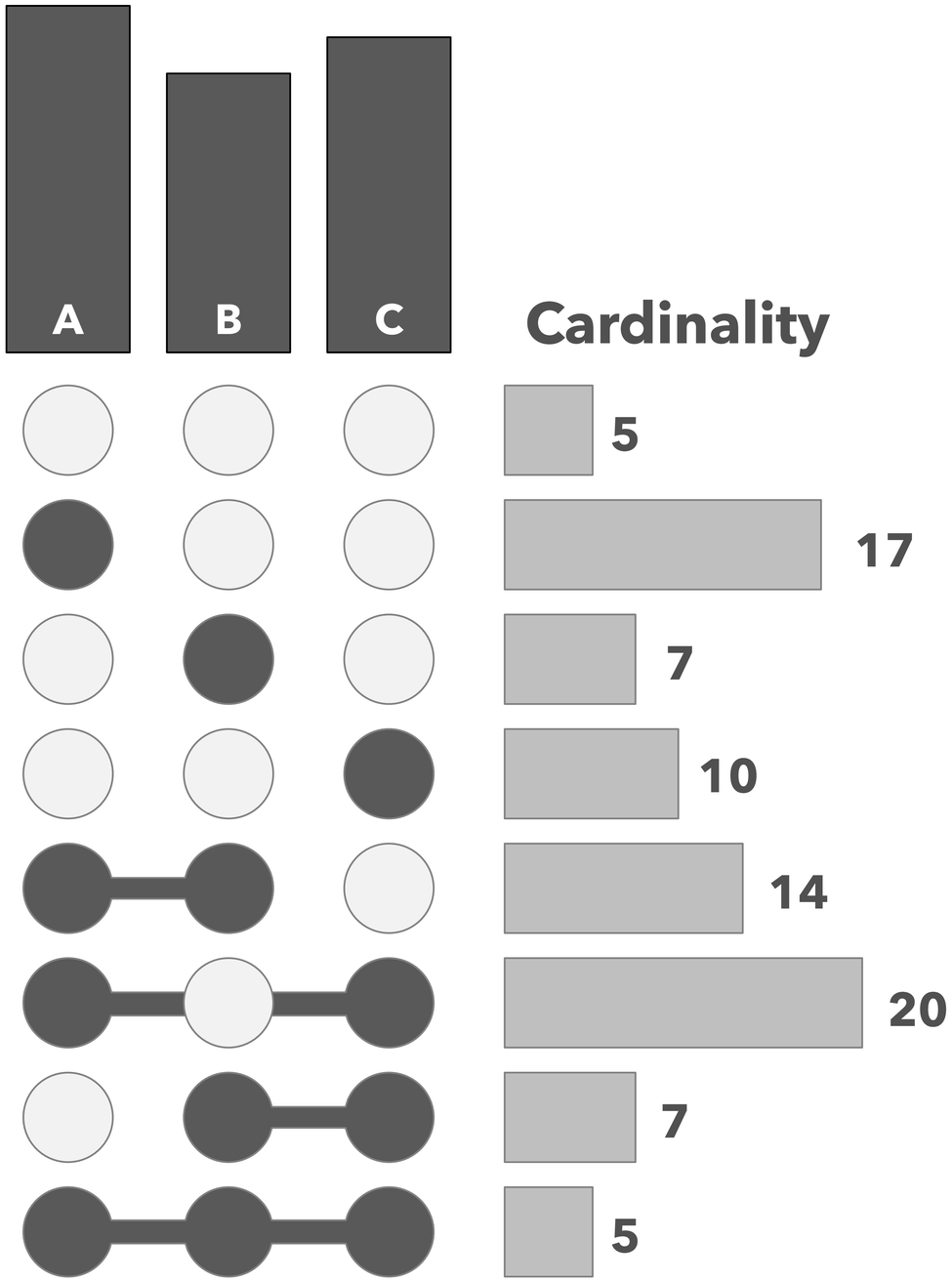

A
B
C

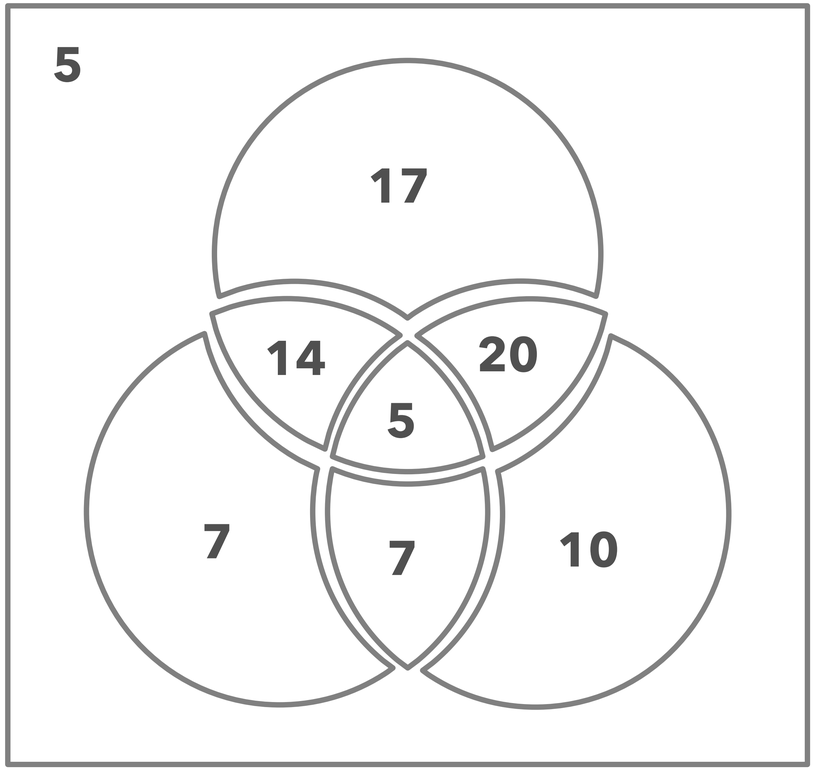

Universal Set
A
B
C



Universal Set

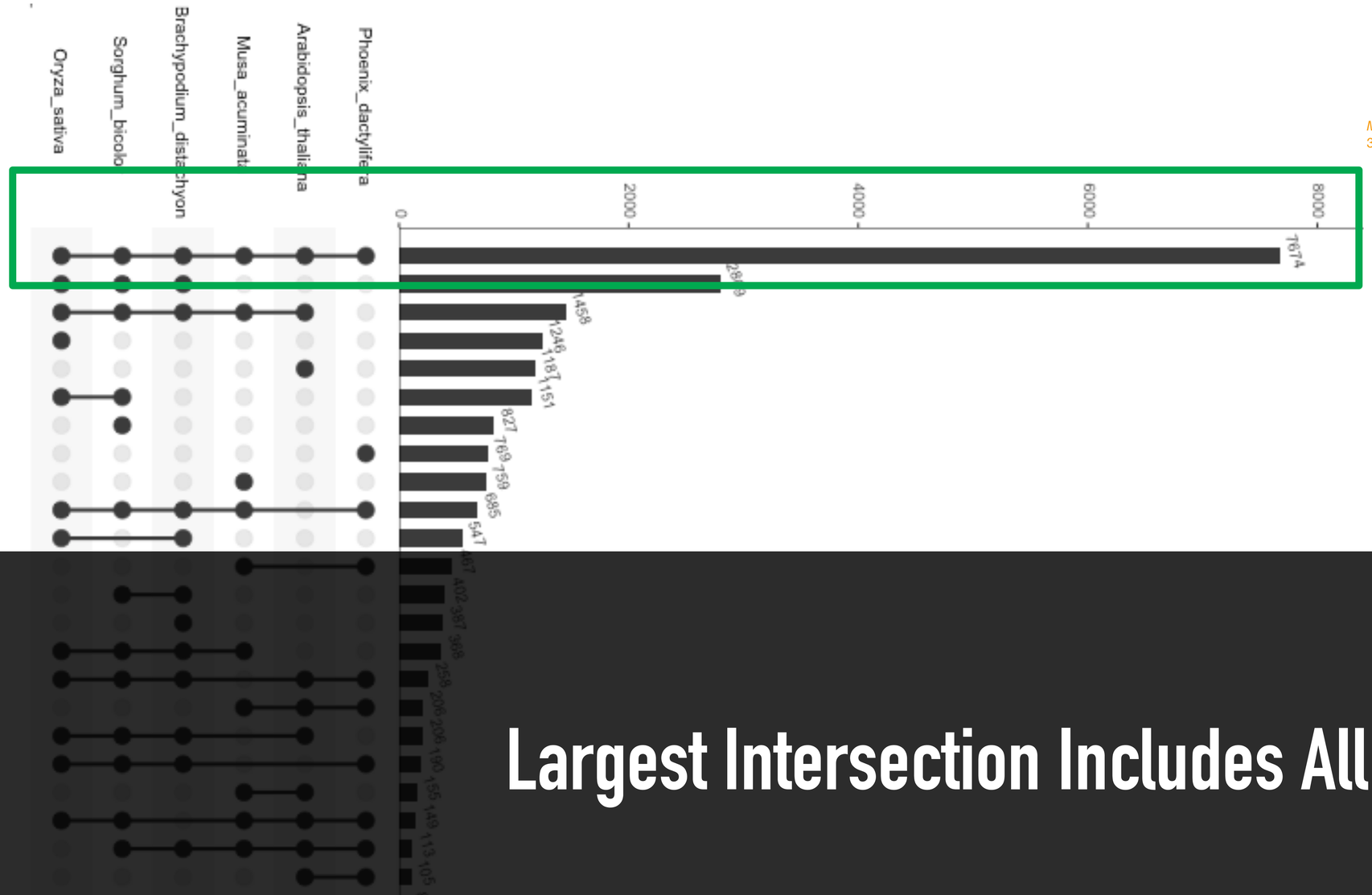


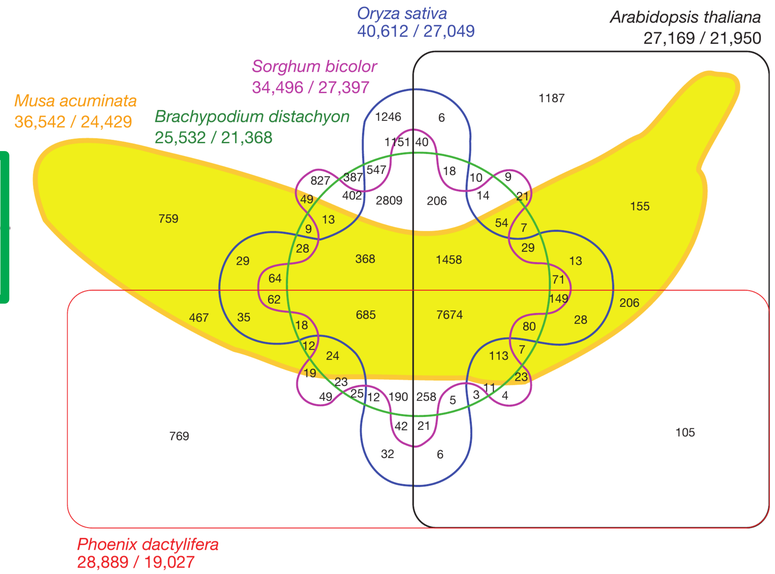

A
B
C
Cardinality
5
17
7
10
14
20
7
5


5
17
14
20
5
7
7
10

https://upset.multinet.app/

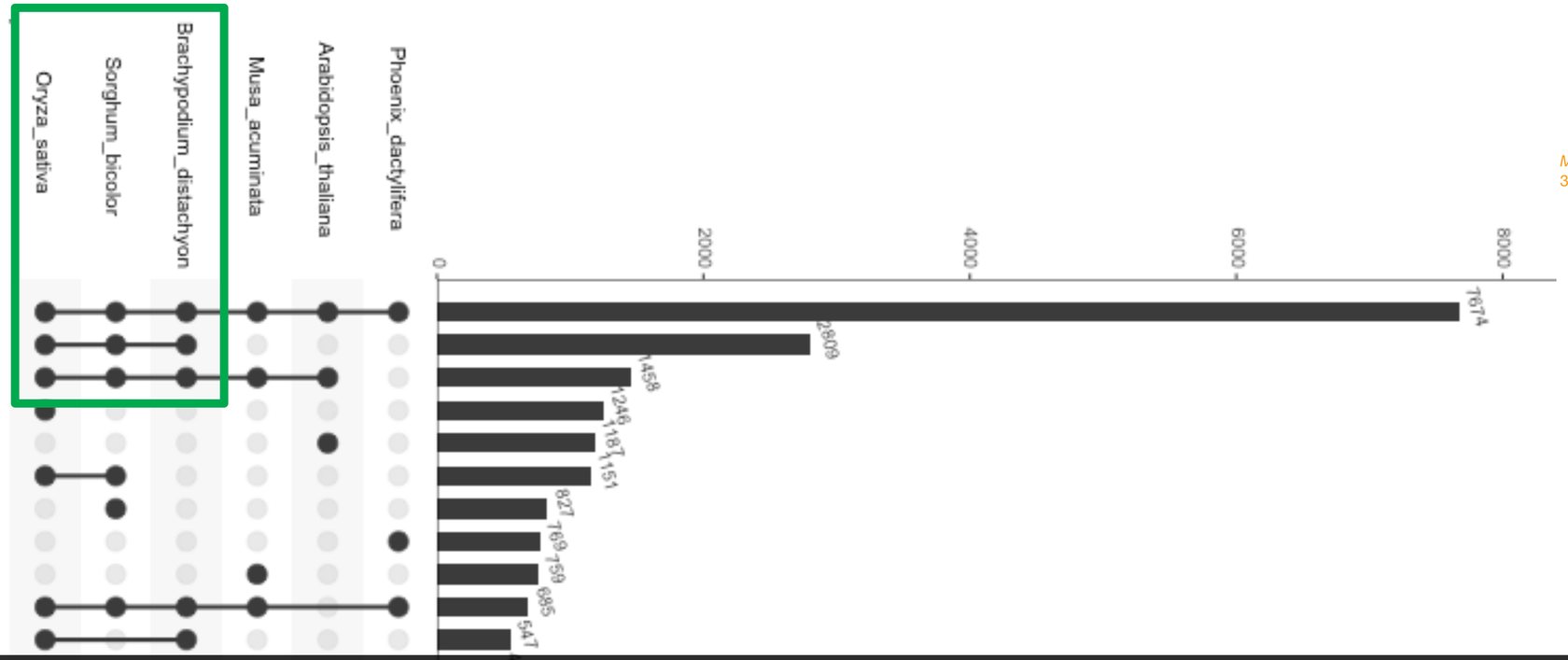
THE BANANA CHART REDESIGNED: UPSET

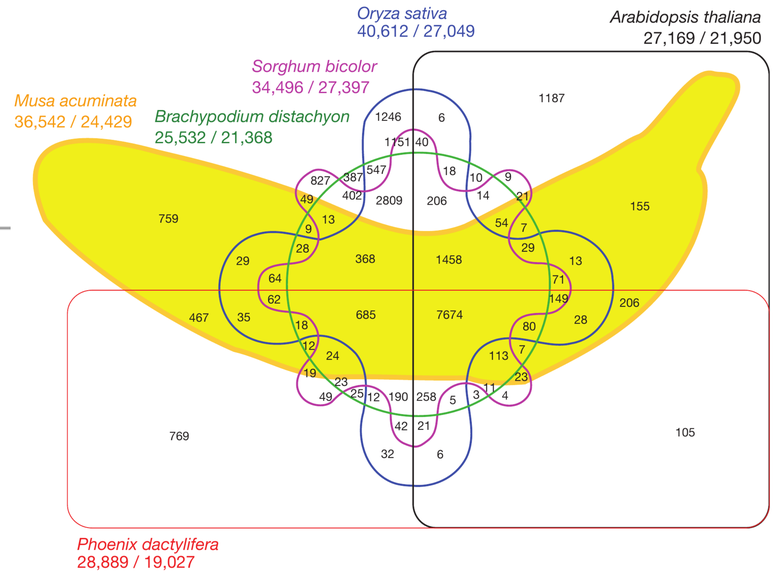




Largest Intersection Includes All Sets

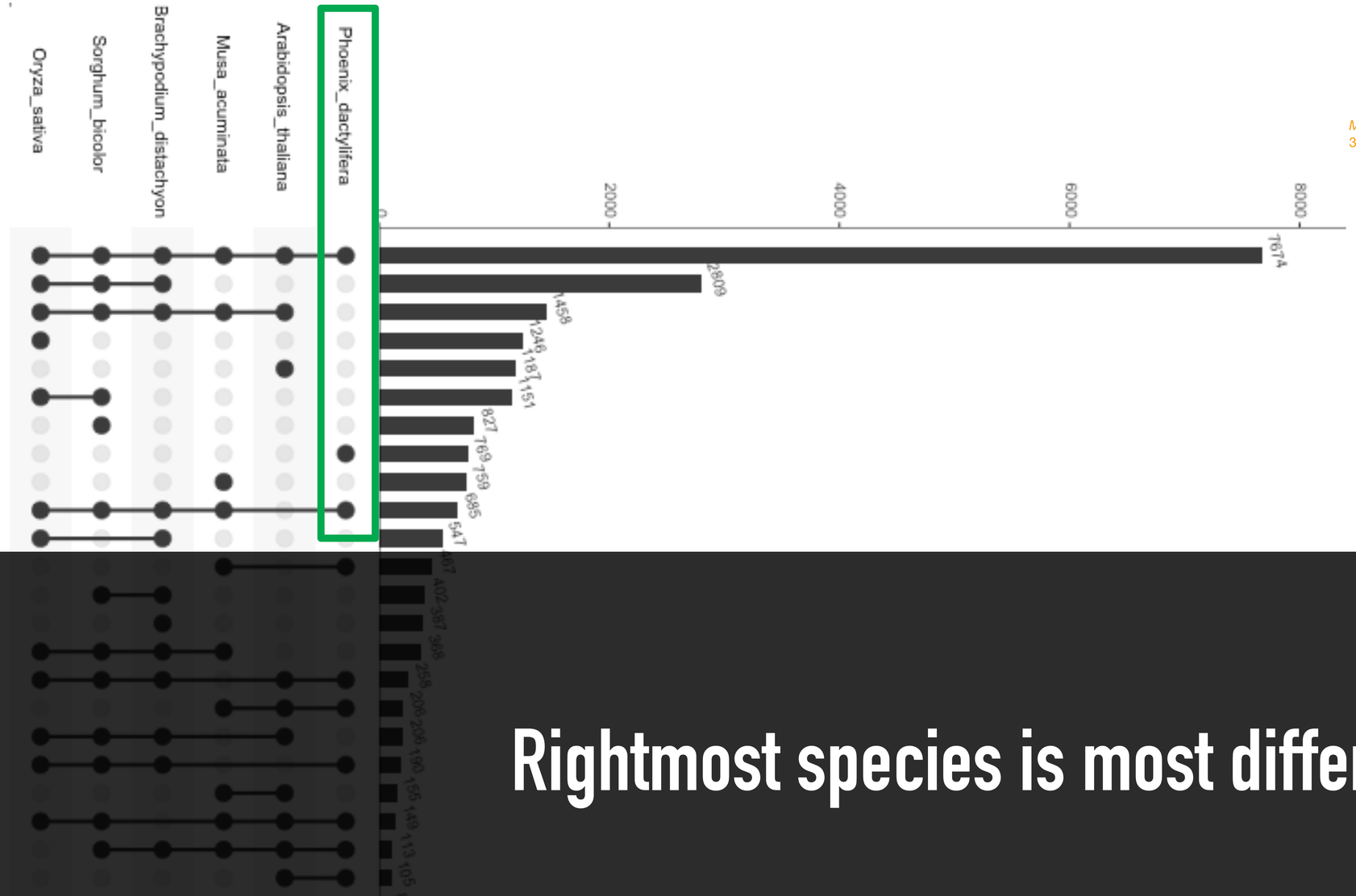
THE BANANA CHART REDESIGNED: UPSET

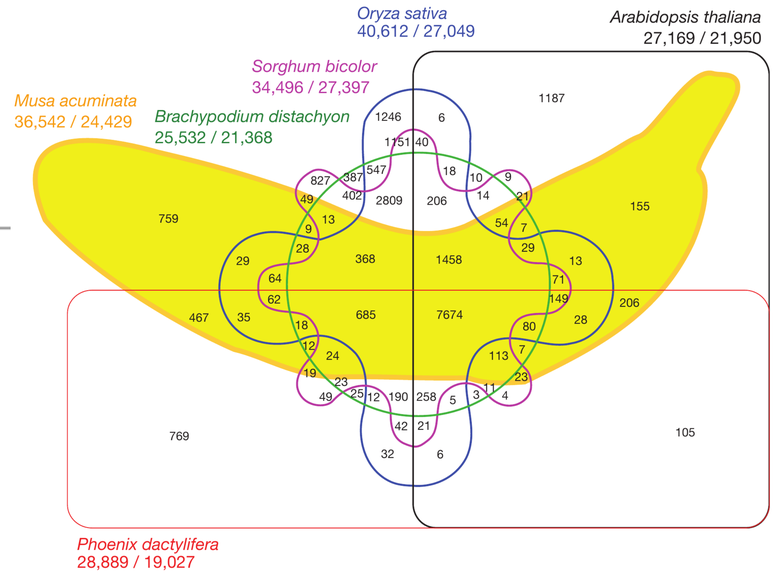




Three Leftmost Species Are Most Similar

THE BANANA CHART REDESIGNED: UPSET





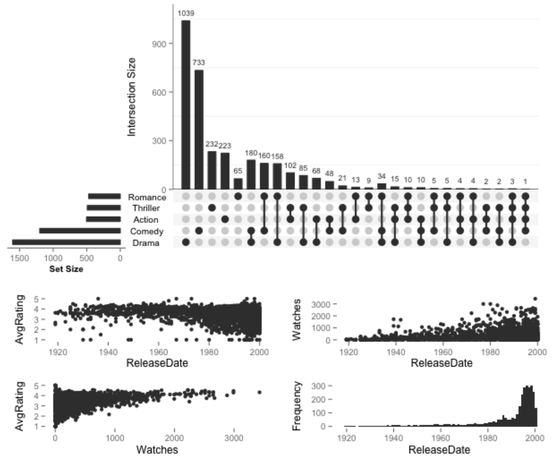
Rightmost species is most different

UPSET

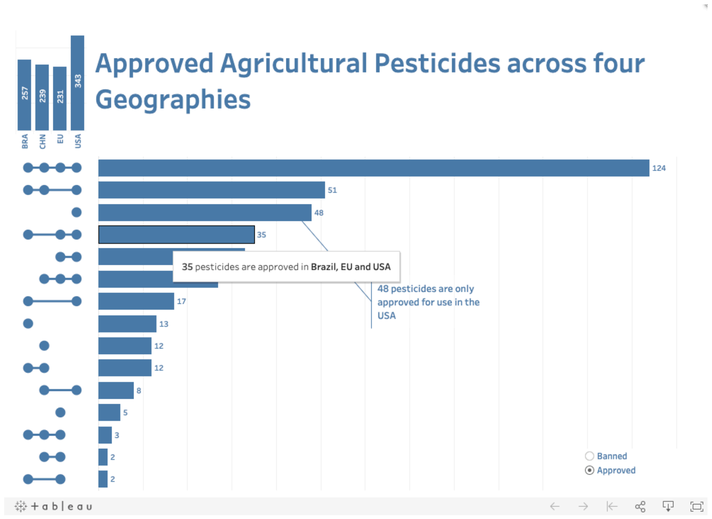
The canonical way to show set data with > 3 sets

Second-most cited VIS paper of the last decade

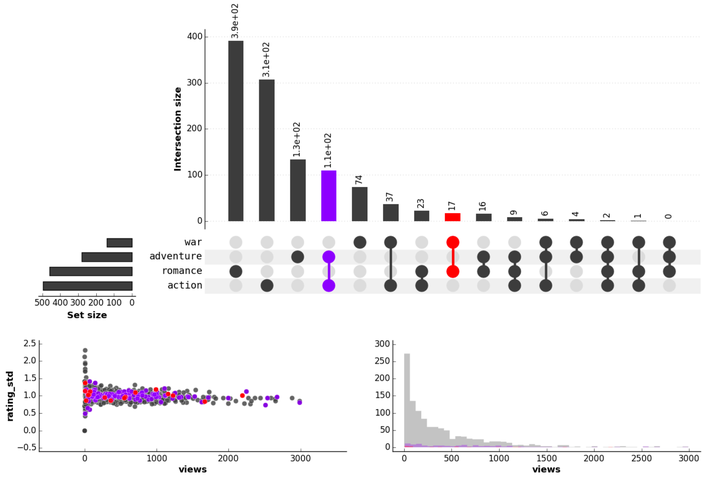
Available in various languages: http://upset.app/



R



Tableau

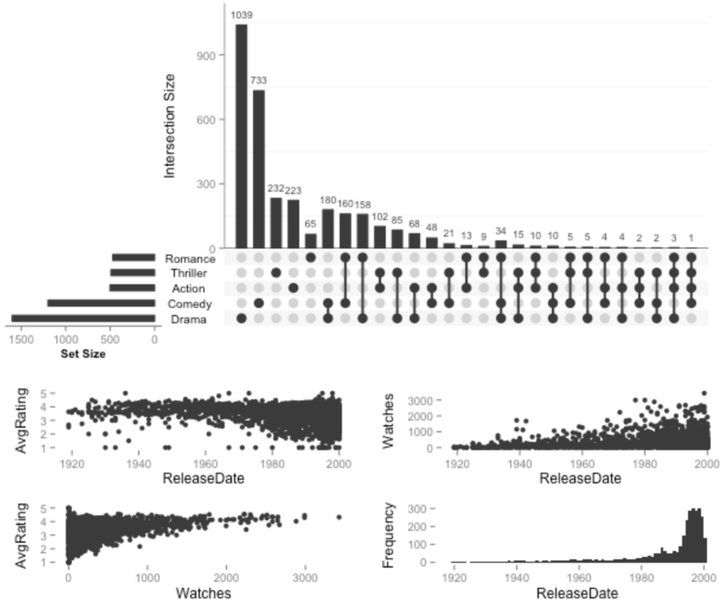


Python

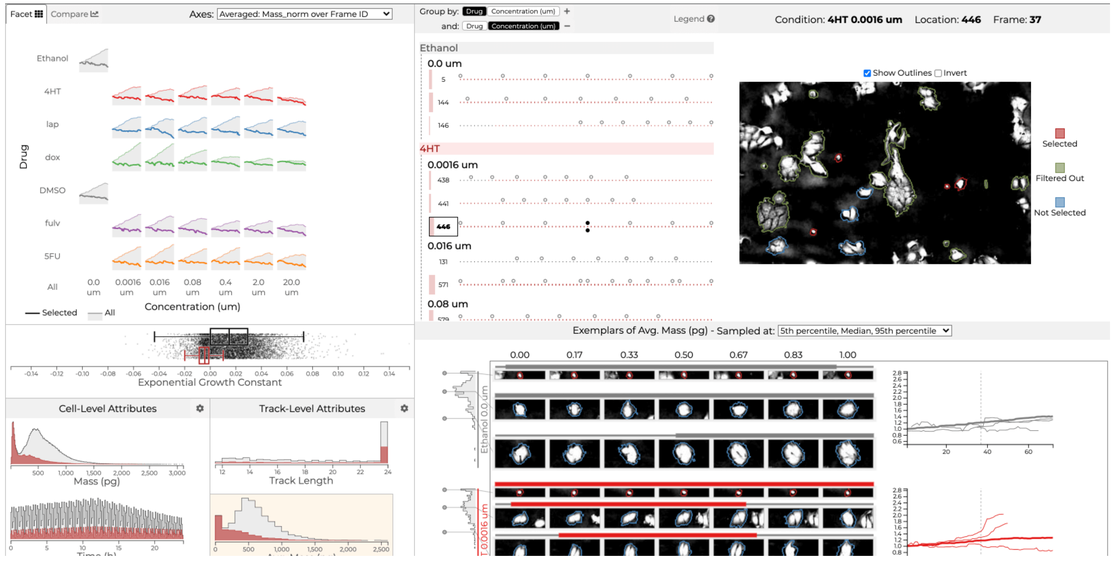
Proper Visualization

Makes a Difference

THE VISUALIZATION SPECTRUM



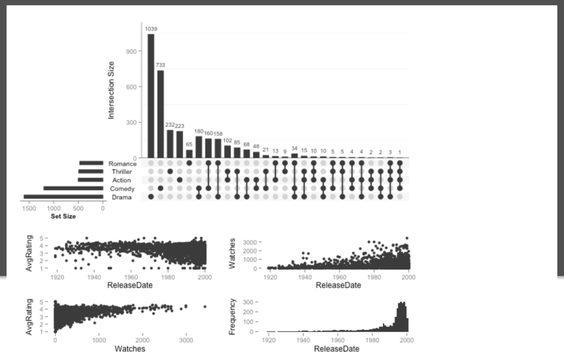

```
upset(movies,attribute.plots=list(gridrows=60,plots=list(list(plot=scatter_plot, x="ReleaseDat
list(plot=scatter_plot, x="ReleaseDate", y="Watches"),list(plot=scatter_plot, x="Watches", y="
list(plot=histogram, x="ReleaseDate")), ncols = 2))
```



Static charts, manually created

Bespoke interactive visualization systems

STATIC CHARTS




```
upset(movies,attribute.plots=list(gridrows=60,plots=list(list(plot=scatter_plot, x="ReleaseDat
list(plot=scatter_plot, x="ReleaseDate", y="Watches"),list(plot=scatter_plot, x="Watches", y="
list(plot=histogram, x="ReleaseDate")), ncols = 2))
```

Pros

Flexibility

Lots of chart types to choose from

Readily available

Good for paper figures

Cons

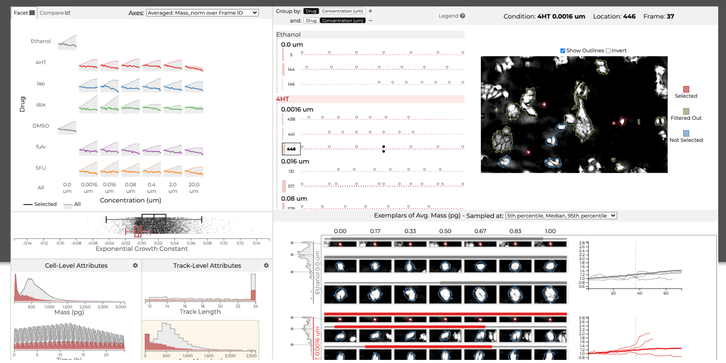
Usually no interaction

Limited to standard charts

Limits for large & complex data

Has to be created by end-user

BESPOKE SYSTEMS



Pros

Useful for complex use cases

Scalable data analysis

Leverages Interaction (search, brushing, sorting, filtering, querying, etc.)

Cons

Difficult / expensive to implement

Difficult to maintain

Requires Technical Partner (vendor, research group)

SO WHAT SHOULD YOU CHOOSE?

REST OF THIS TALK

Part 1: Making “static” charts a bit smarter

Part 2: Loon – an example of a bespoke system for a high stakes application

Subtext: If you’re looking for collaborations anywhere along the spectrum, reach out!

TECHNICAL CONTRIBUTIONS

Kiran Gadhave, Zach Cutler

REPRODUCIBILITY, REUSABILITY, AND INTEGRATION FOR INTERACTIVE VISUALIZATION

[Information Visualization 2021, EuroVis 2022, EuroVis 2024]

BRIDGING BETWEEN DATA ANALYSIS MODALITIES

BRIDGING BETWEEN DATA ANALYSIS MODALITIES

What are Modalities?

1. Interactive Vis

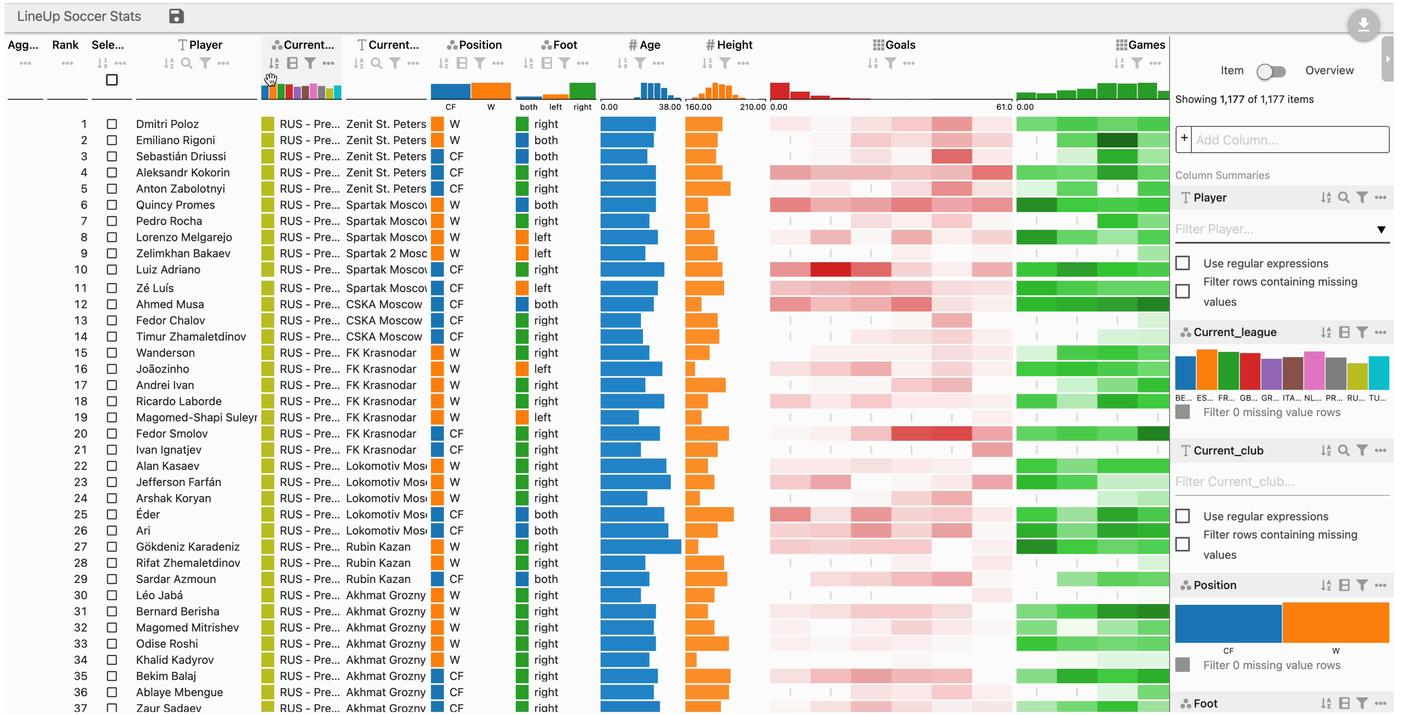
2. Code

INTERACTIVE VISUALIZATION: BENEFITS

Intuitive

Easy to use

Uses human perceptual capabilities



INTERACTIVE VISUALIZATION: DOWNSIDES

Limited Expressivity

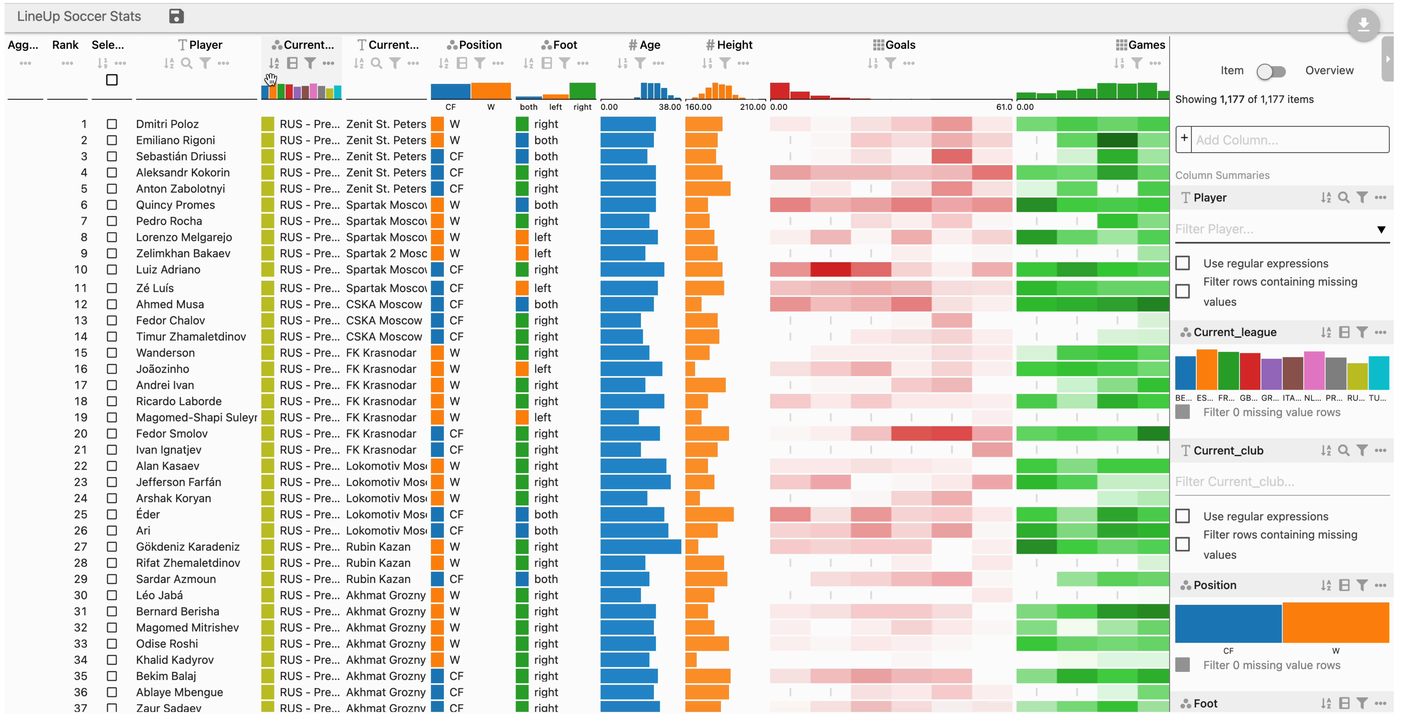
Some operations are difficult

e.g., conditional queries..

Not reusable

need to redo analysis when data changes

Not reproducible



CODE: BENEFITS

Flexible and powerful

you basically can do anything

Reusable

if your data changes, re-run

Reproducible

everything is documented

```
# Keep this cell
avy_df = pd.read_csv('./avalanches.csv')

# Remove NaN coordinates
avy_df = avy_df[avy_df['Coordinates']==avy_df['Coordinates']]

# Split into latitude & longitude
avy_df[['lat', 'lon']] = avy_df['Coordinates'].str.split(',', expand=True)

# Remove values outside of Utah bounds
avy_df = avy_df[  (36 < avy_df['lat'].astype('float')) & (avy_df['lat'].astype('float') <   42)]
avy_df = avy_df[(-114 < avy_df['lon'].astype('float')) & (avy_df['lon'].astype('float') < -108)]

# Keep columns we need
avy_df = avy_df[['Date', 'Region', 'Trigger', 'lat', 'lon']]
avy_df.head()
```

CODE / SCRIPTING: DOWNSIDES

It's hard

requires extensive training

reading documentation

not discoverable

It's time consuming

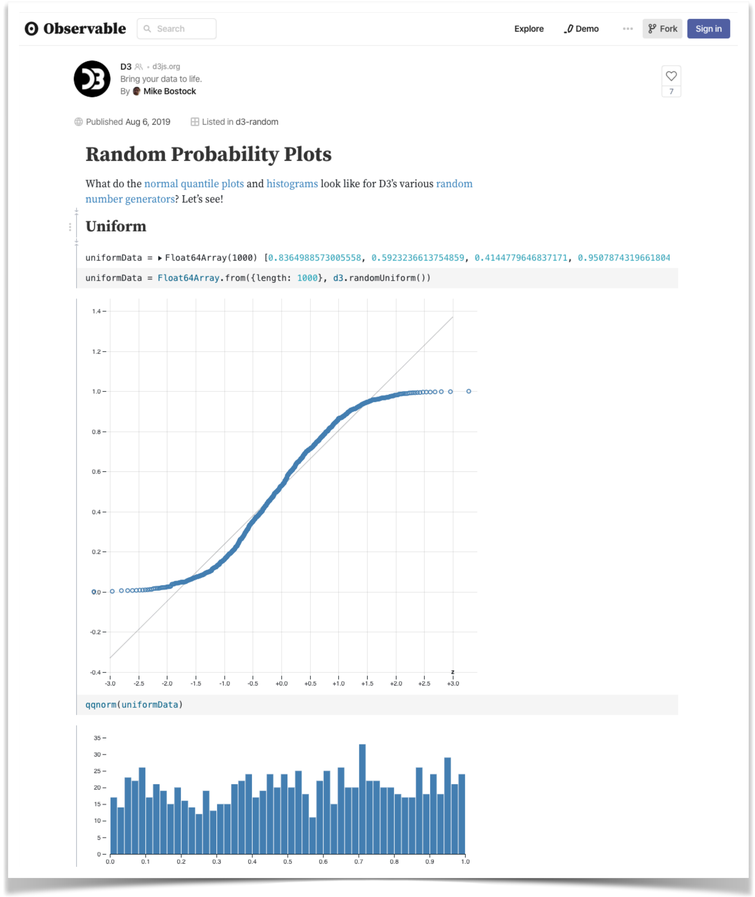
even simple things require effort

Some operations are difficult

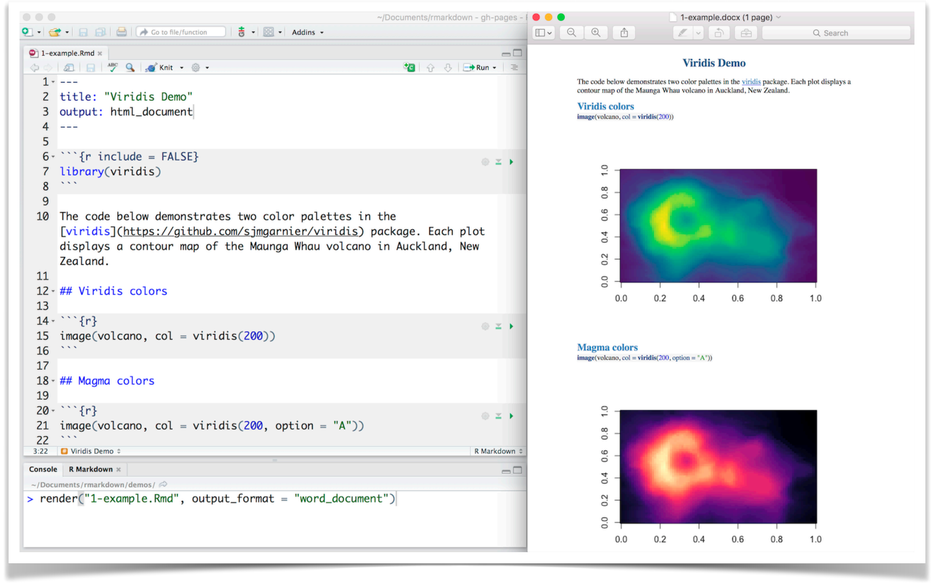
e.g., labeling data points

```
# Keep this cell
avy_df = pd.read_csv('./avalanches.csv')

# Remove NaN coordinates
avy_df = avy_df[avy_df['Coordinates']==avy_df['Coordinates']]

# Split into latitude & longitude
avy_df[['lat', 'lon']] = avy_df['Coordinates'].str.split(',', expand=True)

# Remove values outside of Utah bounds
avy_df = avy_df[  (36 < avy_df['lat'].astype('float')) & (avy_df['lat'].astype('float') <   42)]
avy_df = avy_df[(-114 < avy_df['lon'].astype('float')) & (avy_df['lon'].astype('float') < -108)]

# Keep columns we need
avy_df = avy_df[['Date', 'Region', 'Trigger', 'lat', 'lon']]
avy_df.head()
```

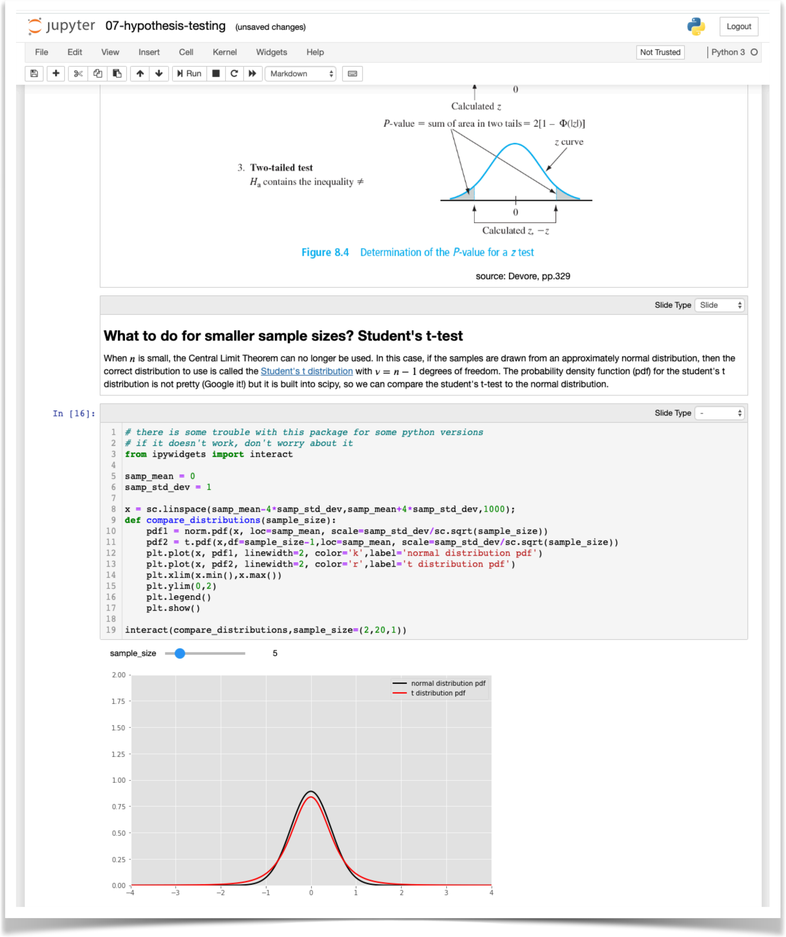
COMPUTATIONAL NOTEBOOKS: A MIDDLE GROUND?



Observable



R Markdown



Jupyter Notebooks

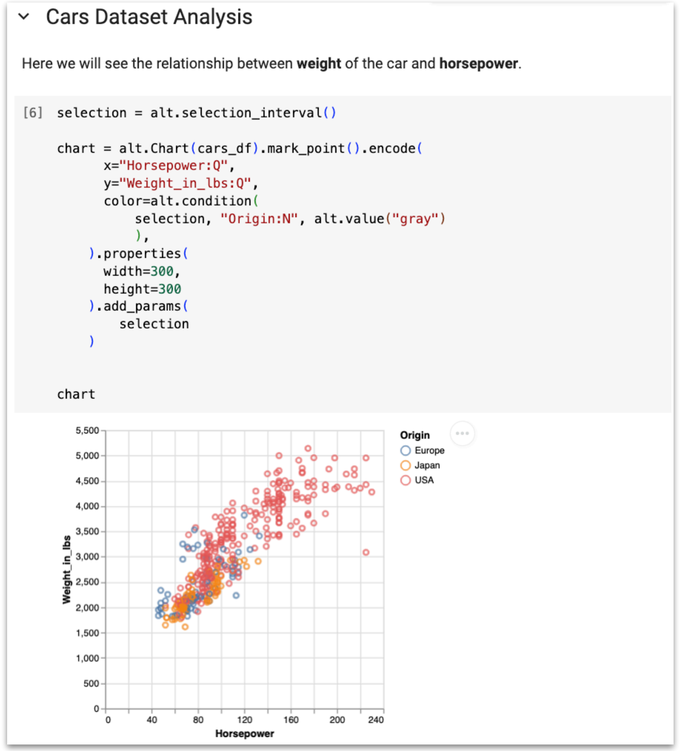
COMPUTATIONAL NOTEBOOKS: A MIDDLE GROUND?

Yes!

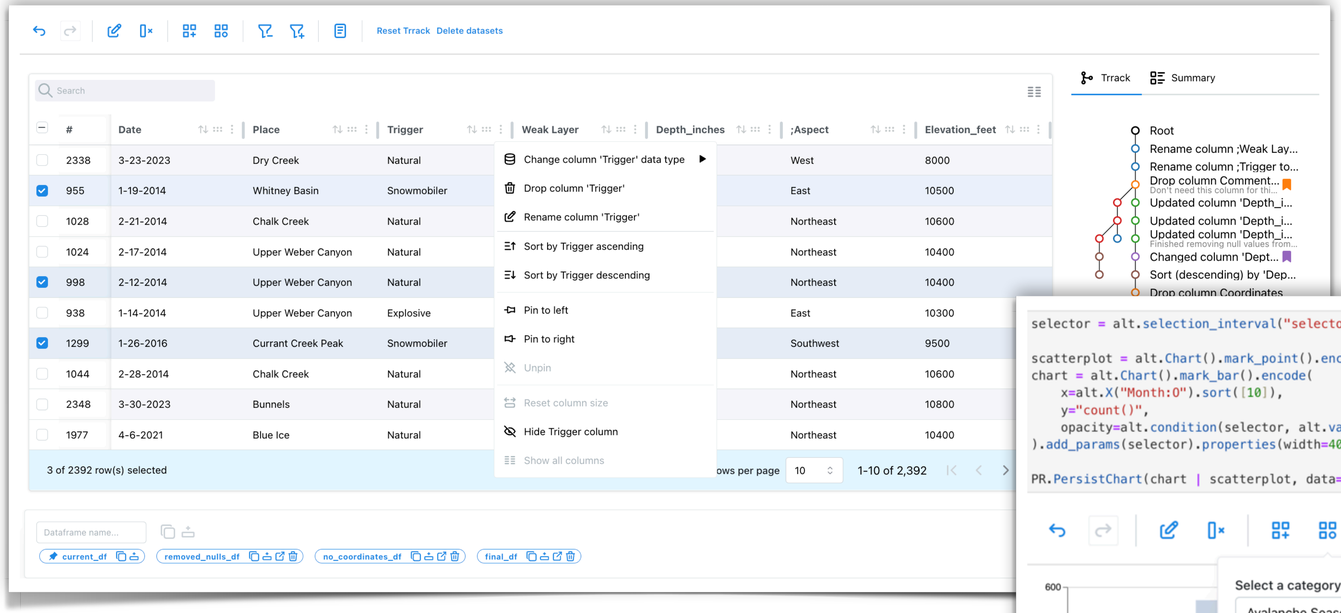
Afford both scripting and interactive visualization

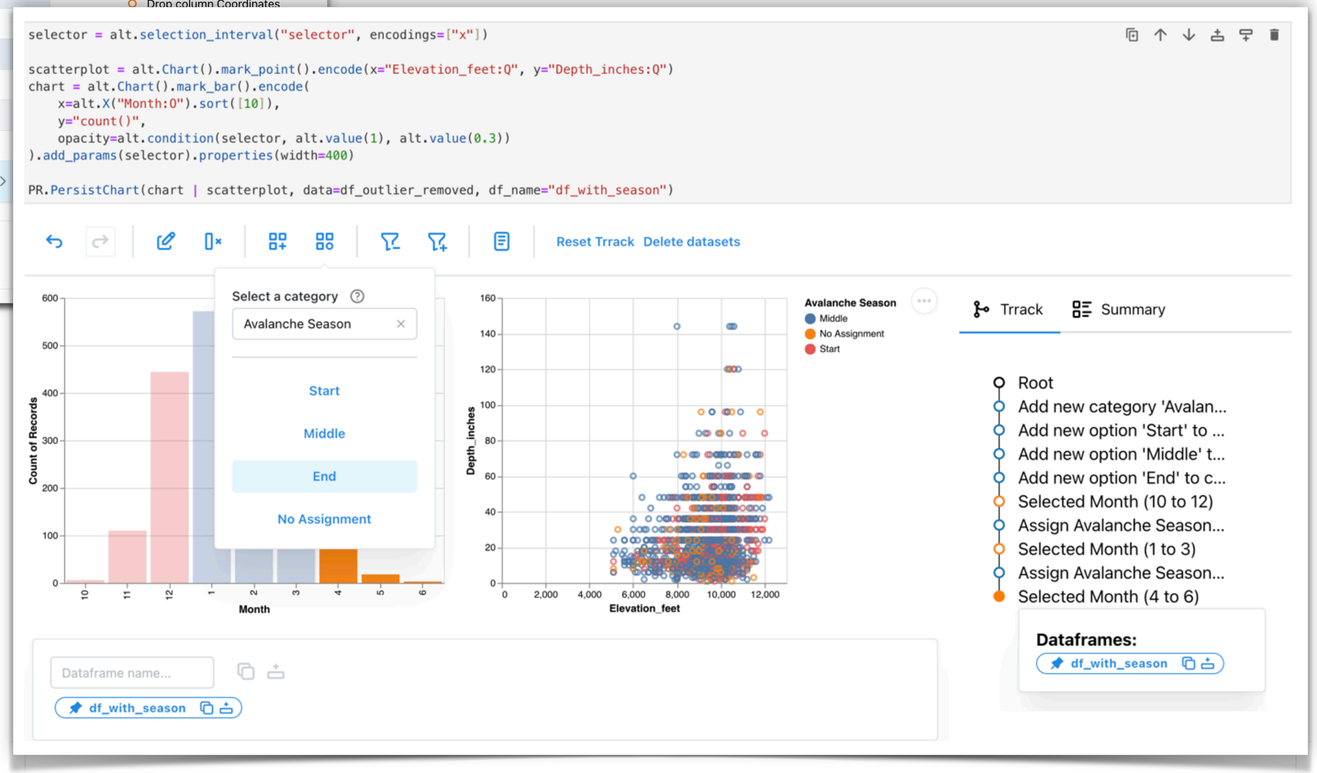
But visualizations are a dead end

- can't "use" interaction in code
- e.g., changing a label, or filtering a value



PERSISTENT AND REUSABLE INTERACTIONS IN COMPUTATIONAL NOTEBOOKS





A Jupyter Plugin

https://vdl.sci.utah.edu/persist/

```
pip install persist_ext
```

Demo Notebook: https://tinyurl.com/5db7nynn

DATASET EXAMPLE: HISTORICAL AVALANCHES IN UTAH

Avalanches are a major hazard

in Utah

Utah Avalanche Center collects data about avalanches, including **where it occurred (location, elevation), how it occurred, how big it was**, etc.

YOU'RE DOING DATA ANALYSIS IN PYTHON

What's the pandas code…

- …to **change the order of columns**?
- …to **drop a column**?
- …to **change the label of a column**?

Nothing here is hard, but it's annoying.

PERSIST MAKES THIS EASY

```
[4]: PR.PersistTable(avalanches, df_name="avalanches")
```

Reset Trrack Delete datasets

Search

#	;Region	Month	Day	Year	;Trigger	;Weak Layer
1	Salt Lake	11	9	2012	Snowboarder	New Snow/Old Snow
2	Salt Lake	11	11	2012	Skier	New Snow/Old Snow
3	Salt Lake	11	11	2012	Skier	Facets
4	Salt Lake	11	11	2012	Skier	New Snow
5	Salt Lake	11	11	2012	Skier	Facets
6	Salt Lake	11	10	2012	Skier	New Snow/Old Snow
7	Salt Lake	11	12	2012	Skier	Facets
8	Salt Lake	12	8	2012	Skier	Facets
9	Salt Lake	12	9	2012	Skier	Facets
10	Salt Lake	12	10	2012	Skier	Facets

Rows per page 10 1-10 of 2,392

Trrack Summary

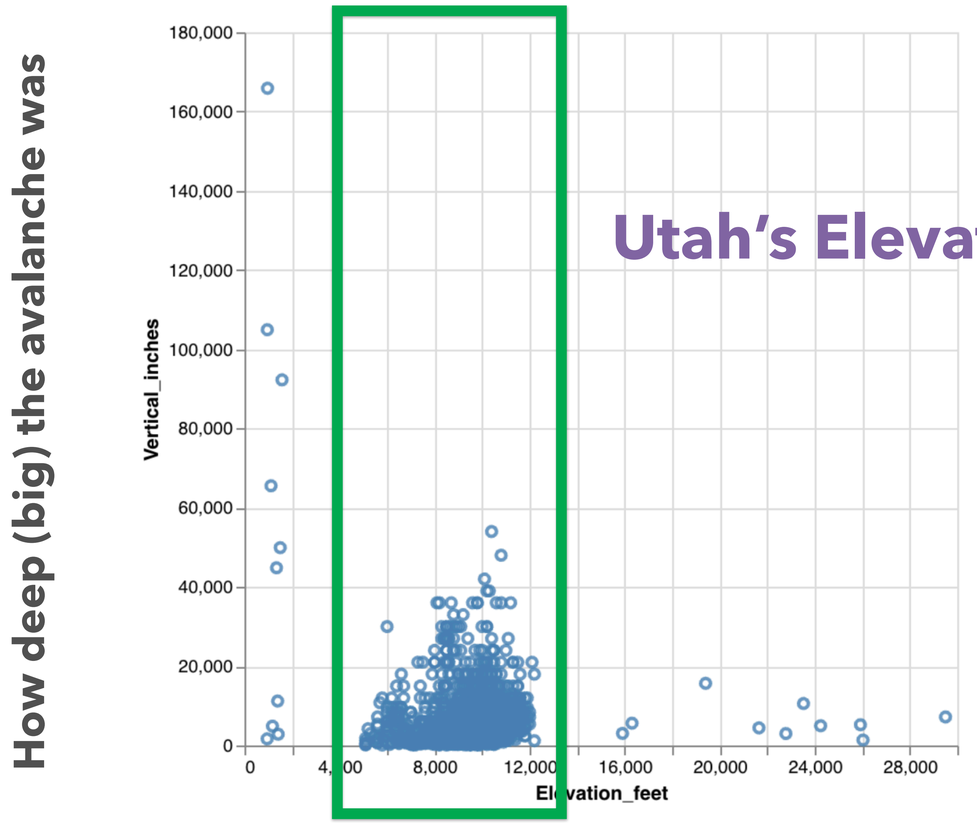
Root

Dataframe name...

avalanches

WHAT IS THIS TALK ABOUT?

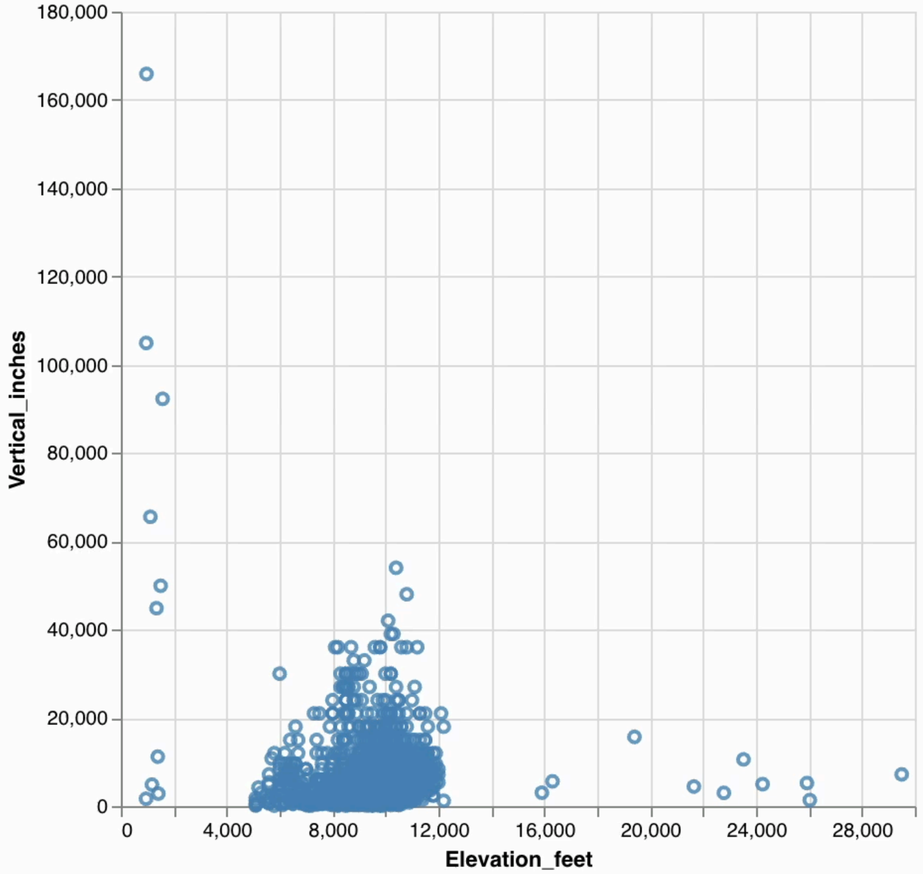
Have you ever plotted something and wished you could just “fix” things as you spot them?



PERSIST MAKES THIS EASY

```
[6]: PR.plot.scatterplot(avalanches, "Elevation_feet:Q", "Vertical_inches:Q", df_name="avalanches")
```

Reset Trrack Delete datasets



Trrack Summary

Root

SO WHAT'S SPECIAL HERE?

Lots of vis tools support these operations

Most data wrangling happens in code: it's just more powerful

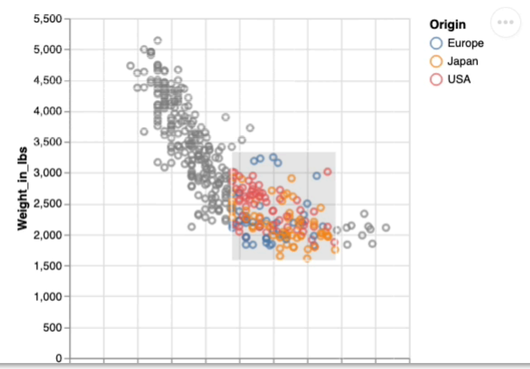
Opportunity: bring interactive operations to code!

Persist works **INSIDE your Jupyter Notebook**

GAPS BETWEEN CODE AND INTERACTIVE OUTPUTS*

Semantic Gap

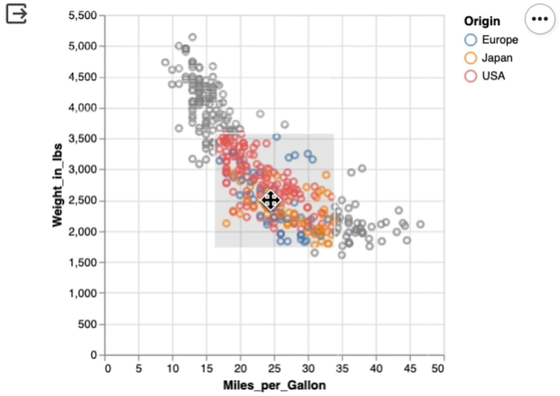
```
selection = alt.selection_interval()

alt.Chart(cars_df).mark_point().encode(
      x="Miles_per_Gallon:Q",
      y="Weight_in_lbs:Q",
      color=alt.condition(
          selection, "Origin:N", alt.value("gray")
          ),
    ).properties(
      width=300,
      height=300
    ).add_params(
        selection
    )

selection = ???
```



Information only flows from code to visualization

Temporal Gap

```
selection = alt.selection_interval()

alt.Chart(cars_df).mark_point().encode(
      x="Miles_per_Gallon:Q",
      y="Weight_in_lbs:Q",
      color=alt.condition(
          selection, "Origin:N", alt.value("gray")
          ),
    ).properties(
      width=300,
      height=300
    ).add_params(
        selection
    )
```



Changes made to code are preserved

Changes made to vis are lost

Layout Gap

```
selection = alt.selection_interval()

# chart = alt.Chart(cars_df).mark_point().encode(
#       x="Miles_per_Gallon:Q",
#       y="Weight_in_lbs:Q",
#       color=alt.condition(
#           selection, "Origin:N", alt.value("gray")
#           ),
#     ).properties(
#       width=300,
#       height=300
#     ).add_params(
#         selection
#     )

chart = alt.Chart(cars_df).mark_point().encode(
      x="Miles_per_Gallon:Q",
      y="Weight_in_lbs:Q",
      color=alt.condition(
          selection, "Origin:N", alt.value("gray")
          ),
    ).properties(
      width=300,
      height=300
    ).add_params(
        selection
    )

chart
```

Changes in code are messy

*[Wu, Hellerstein, Satyanarayan, UIST 2020]

THESIS: BRIDGING BETWEEN CODE AND INTERACTIVE VIS IS USEFUL

Easy handoffs are important!

THE PERSIT APPROACH

PRINCIPLE

Track events in interactive visualizations

Map them to data frame operations

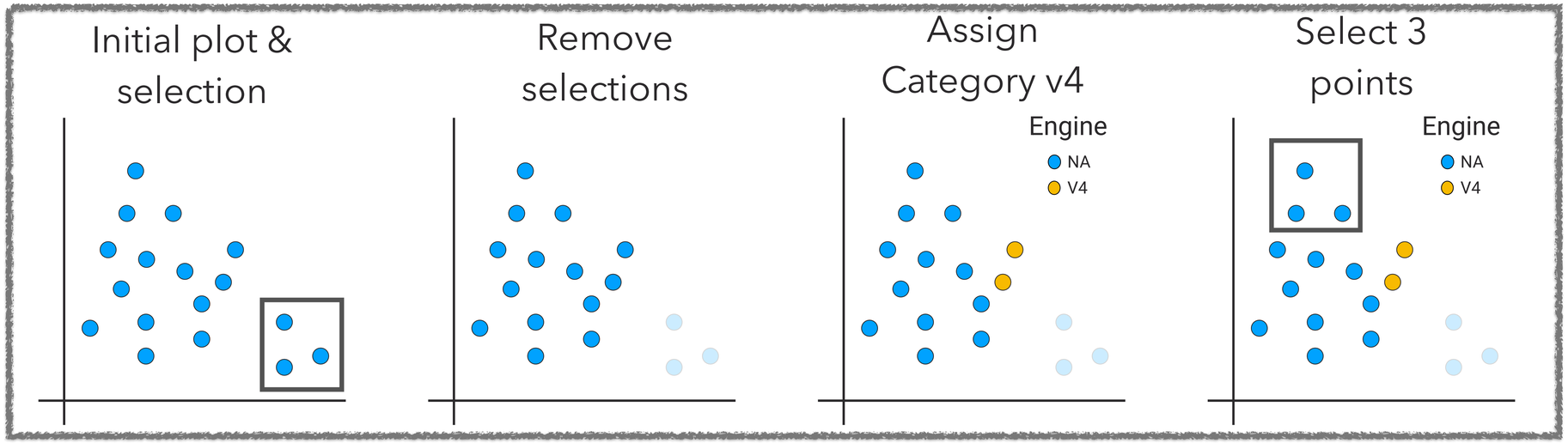
Operations then applied to data frame

HOW IT WORKS

Code to create chart

```
df = pd.read_csv('cars.csv')

PersistChart(scatterplot(df))
```

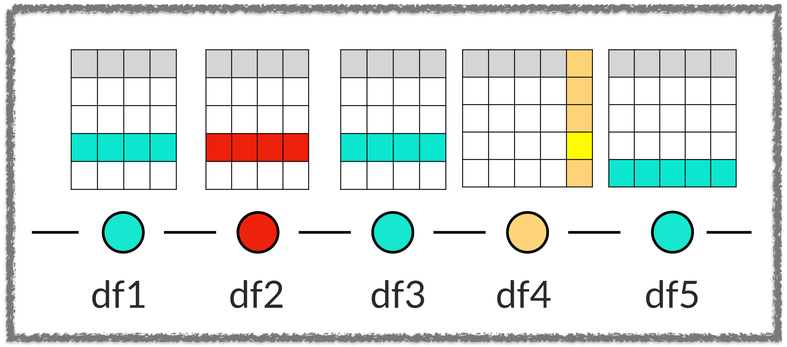
Interactive Visualization



Provenance

- df1 — Select 3 points
- df2 — Remove selections
- df3 — Select 2 points
- df4 — Assign category v4 to selection
- **df5** — Select 3 points

Dataframe updates



In code

```
df5.head()
```

Name	MPG	Cylinders	origin	Engine	is_selected
ford	18	8	USA	NA	FALSE
dodge	15	8	USA	NA	FALSE
volkswagen	22	4	Europe	V4	FALSE
amc	16	8	USA	NA	TRUE

27 rows x 8 columns

OPERATIONS

- Selection
- Edit column names, edit cells
- Sort rows/columns
- Drop columns
- Filter (in/out) items
- Label items
- Categorize items
- Change data types

PERSIST WORKFLOW

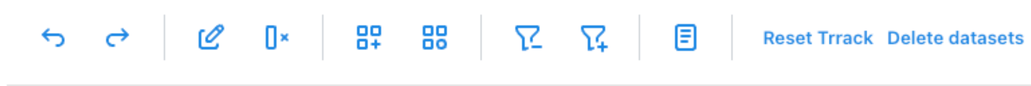
Standard Vega-Altair Chart

```python
import altair as alt
from vega_datasets import data
import persist_ext as PR

cars_df = data.cars() # Load cars dataset

brush = alt.selection_interval() # Create a 2d brush

# create scatterplot and link to brush
scatterplot = alt.Chart(cars_df).mark_point().encode(
    x="Miles_per_Gallon:Q",
    y="Weight_in_lbs:Q",
    color=alt.condition(brush, alt.value("steelblue"), alt.value("gray"))
).add_params(
    brush
)

PR.PersistChart(scatterplot)
```

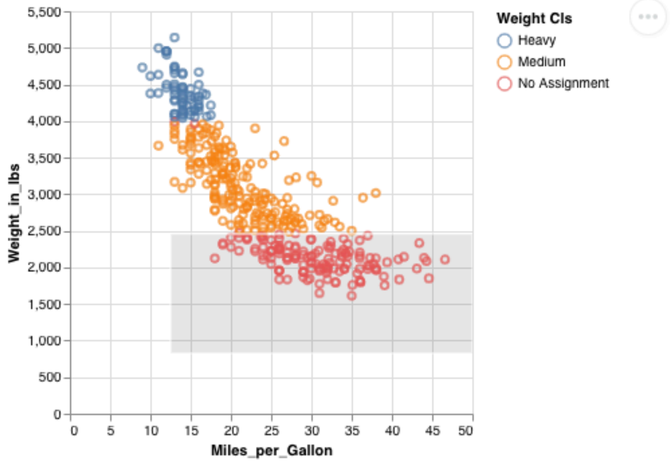
Call to Persist with Chart as Parameter

```python
import altair as alt
from vega_datasets import data
import persist_ext as PR

cars_df = data.cars() # Load cars dataset

brush = alt.selection_interval() # Create a 2d brush

# create scatterplot and link to brush
scatterplot = alt.Chart(cars_df).mark_point().encode(
    x="Miles_per_Gallon:Q",
    y="Weight_in_lbs:Q",
    color=alt.condition(brush, alt.value("steelblue"), alt.value("gray"))
).add_params(
    brush
)

PR.PersistChart(scatterplot)
```

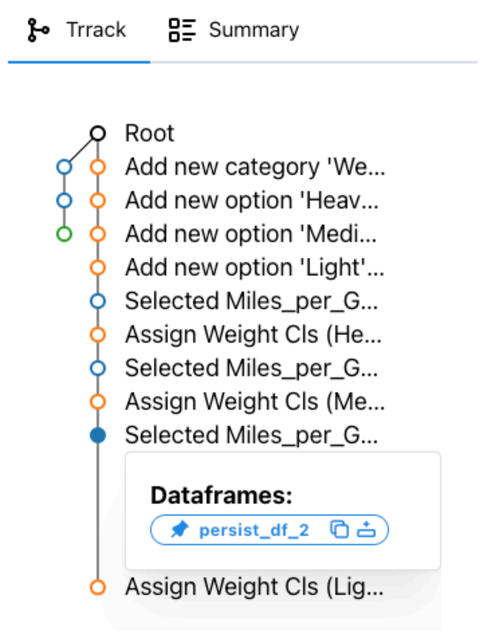
Persist Toolbar



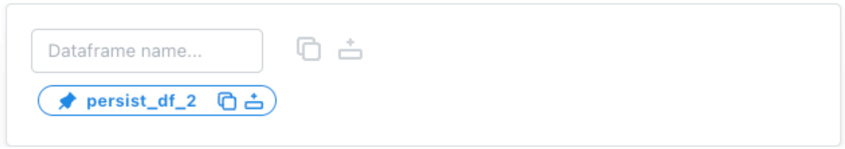
Vega-Altair Chart



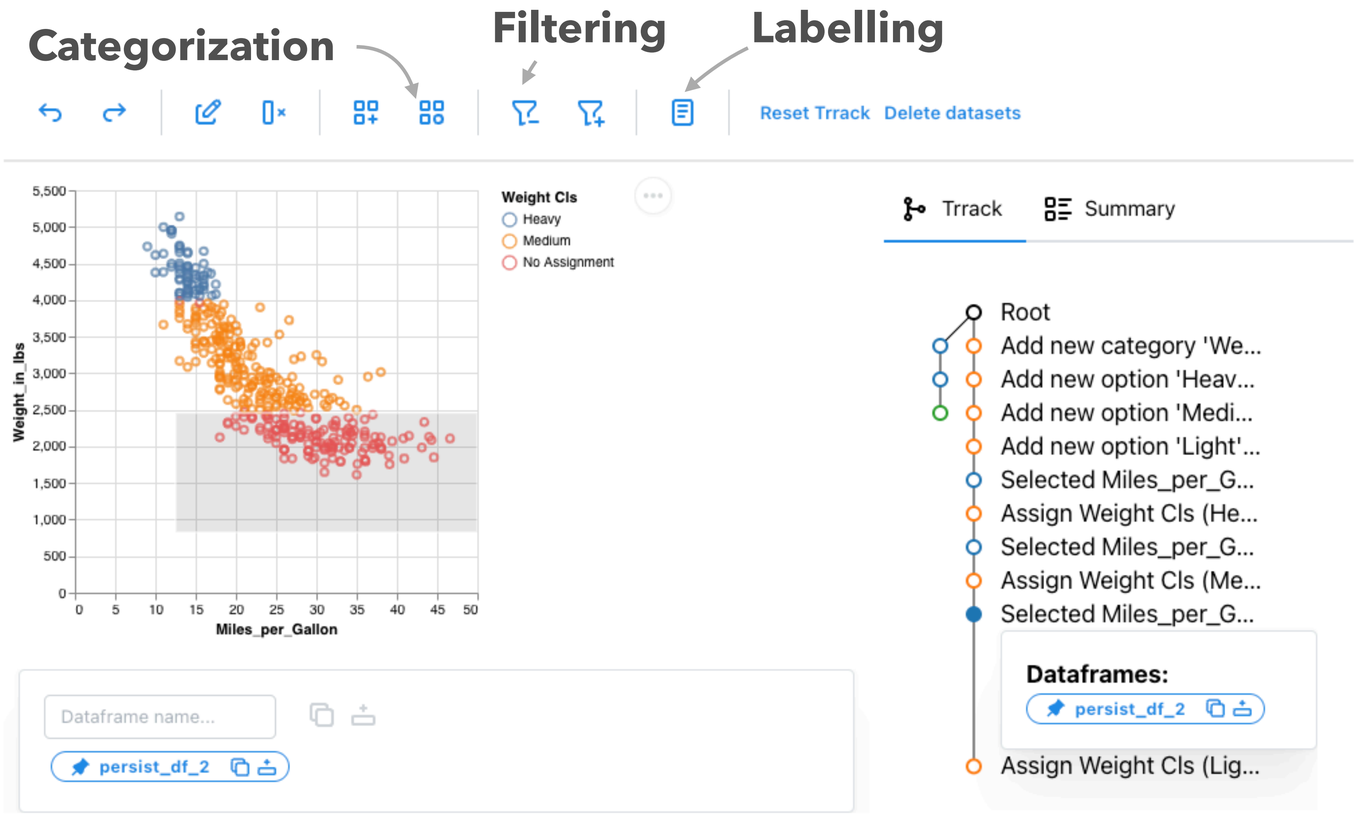
Provenance History



Dataframe Manager

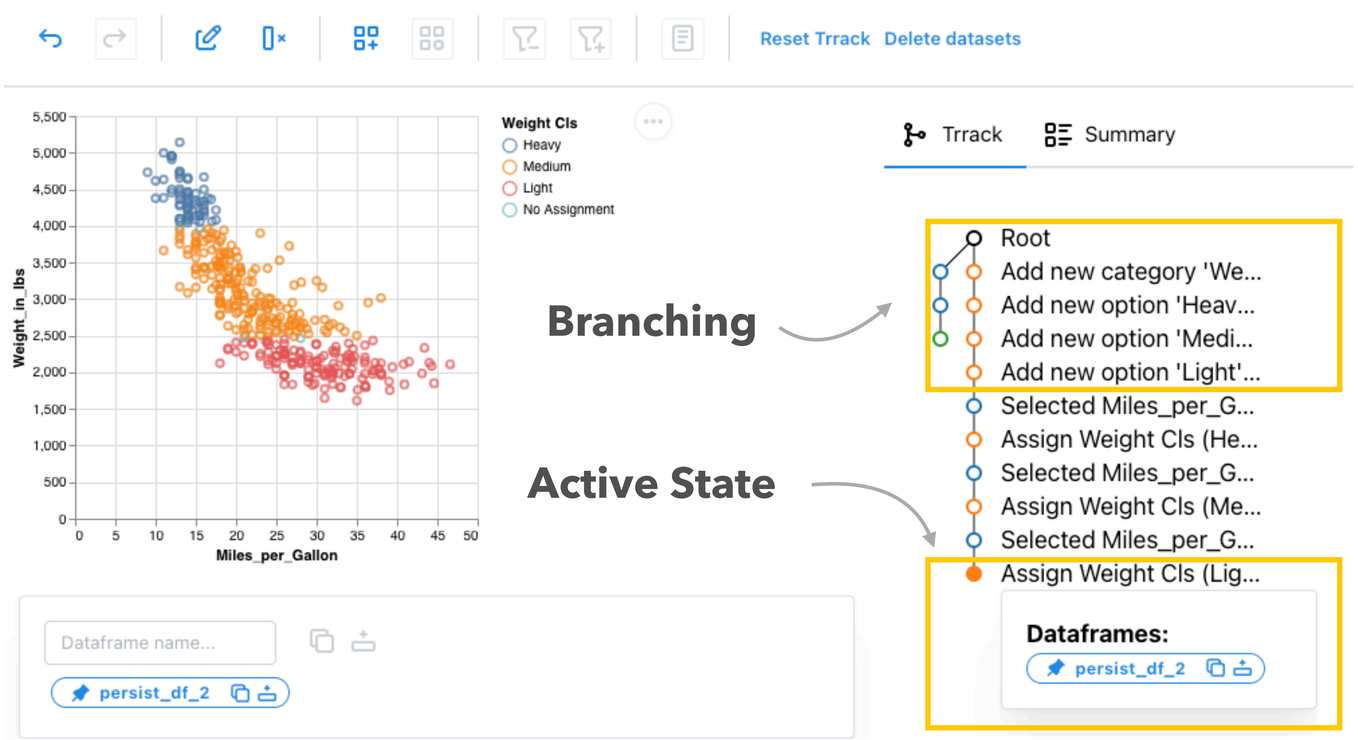


```python
import altair as alt
from vega_datasets import data
import persist_ext as PR

cars_df = data.cars() # Load cars dataset

brush = alt.selection_interval() # Create a 2d brush

# create scatterplot and link to brush
scatterplot = alt.Chart(cars_df).mark_point().encode(
    x="Miles_per_Gallon:Q",
    y="Weight_in_lbs:Q",
    color=alt.condition(brush, alt.value("steelblue"), alt.value("gray"))
).add_params(
    brush
)

PR.PersistChart(scatterplot)
```



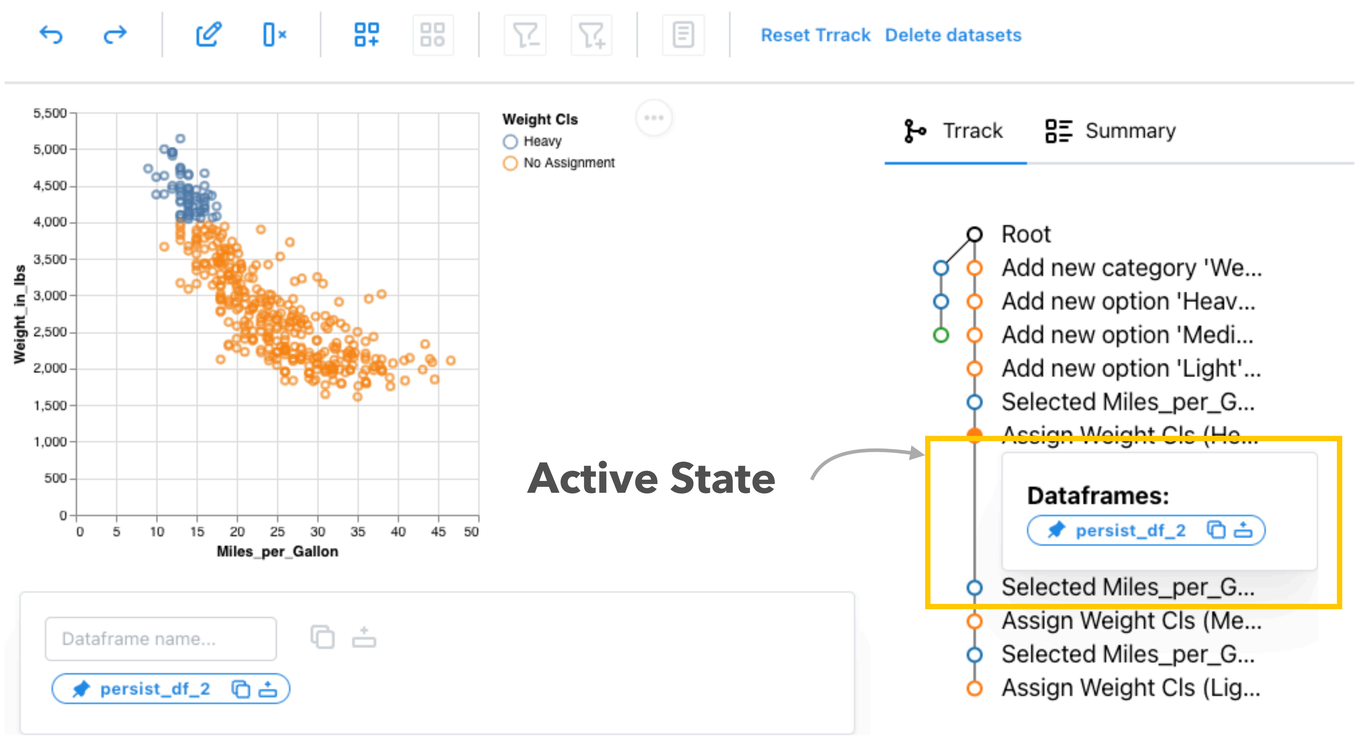
Branches and **Choosing a State** in provenance support *non-linear analysis*, addressing **the layout gap**

```python
import altair as alt
from vega_datasets import data
import persist_ext as PR

cars_df = data.cars() # Load cars dataset

brush = alt.selection_interval() # Create a 2d brush

# create scatterplot and link to brush
scatterplot = alt.Chart(cars_df).mark_point().encode(
    x="Miles_per_Gallon:Q",
    y="Weight_in_lbs:Q",
    color=alt.condition(brush, alt.value("steelblue"), alt.value("gray"))
).add_params(
    brush
)

PR.PersistChart(scatterplot)
```

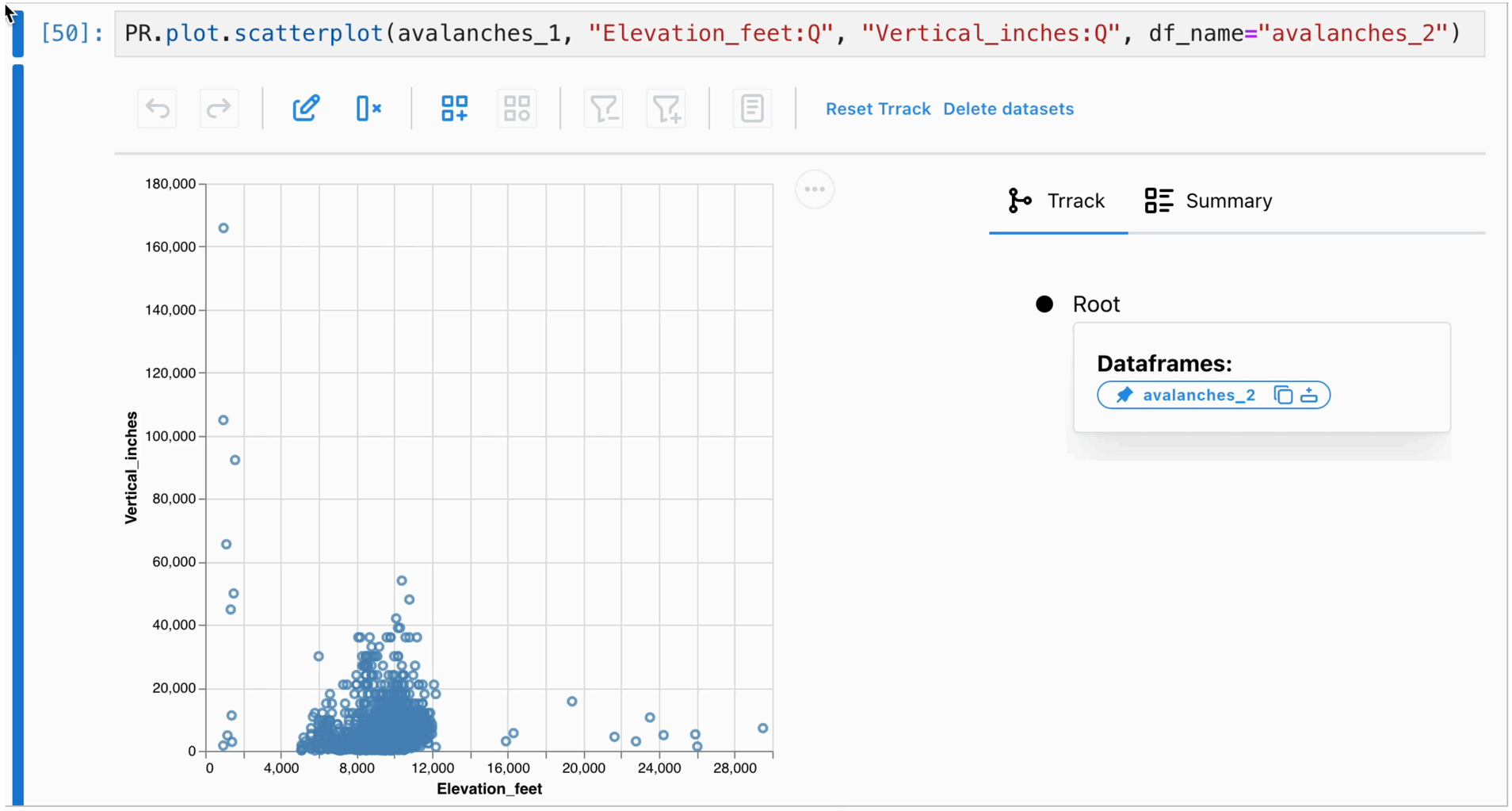


Branches and **Choosing a State** in provenance support *non-linear analysis*, addressing **the layout gap**

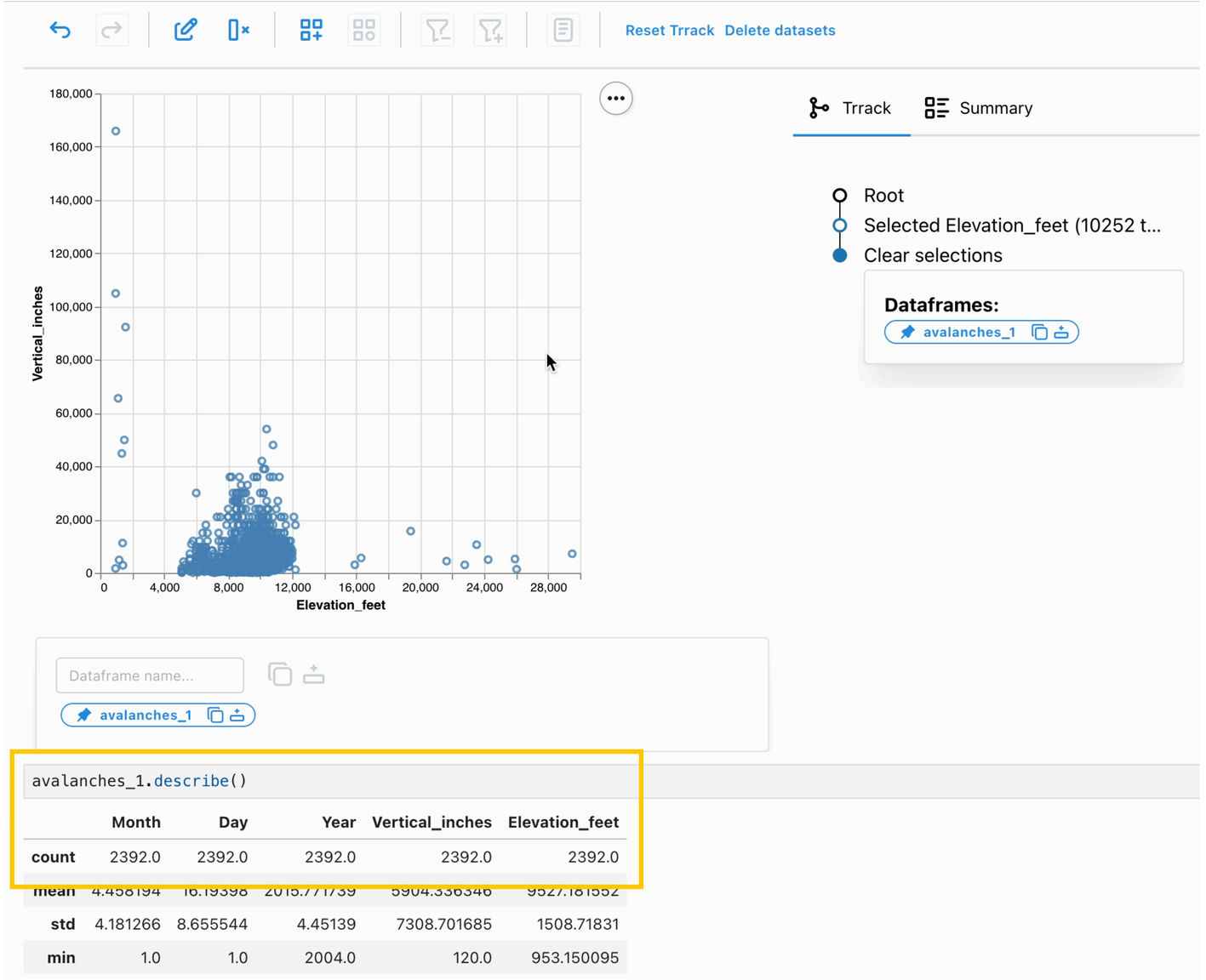
```python
import altair as alt
from vega_datasets import data
import persist_ext as PR

cars_df = data.cars() # Load cars dataset

brush = alt.selection_interval() # Create a 2d brush

# create scatterplot and link to brush
scatterplot = alt.Chart(cars_df).mark_point().encode(
    x="Miles_per_Gallon:Q",
    y="Weight_in_lbs:Q",
    color=alt.condition(brush, alt.value("steelblue"), alt.value("gray"))
).add_params(
    brush
)

PR.PersistChart(scatterplot)
```



Persist **re-runs the interactions** in the output, addressing **the temporal gap**



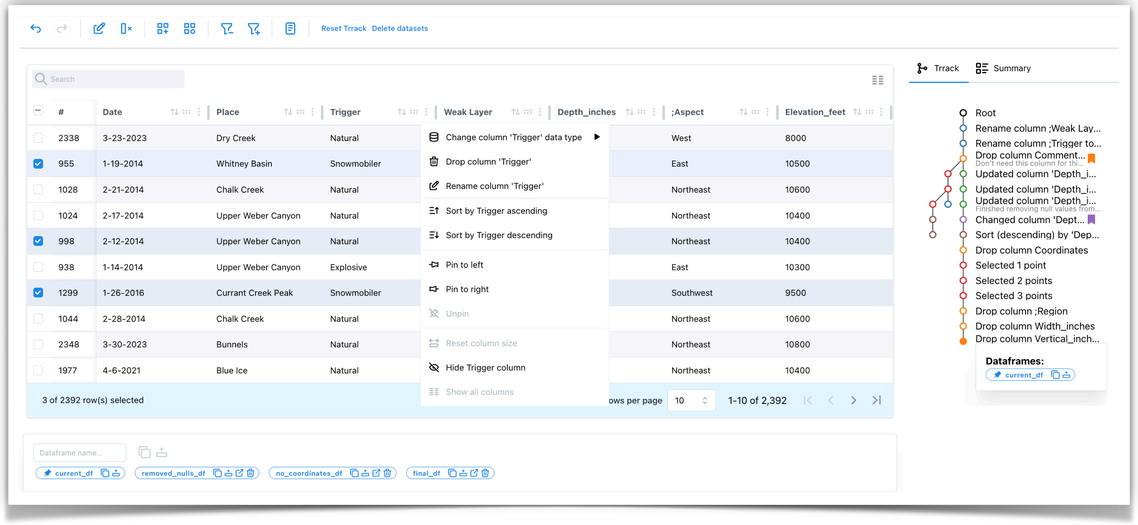
Persist *applies interactions to data frames* that can be accessed in code, addressing the **semantic gap**



Manipulated data frame accessed in code

`count: 2392→79`

VISUALIZATION OPTIONS



Arbitrary Vega-Altair Charts

An Interactive Data Table

VEGA-ALTAIR

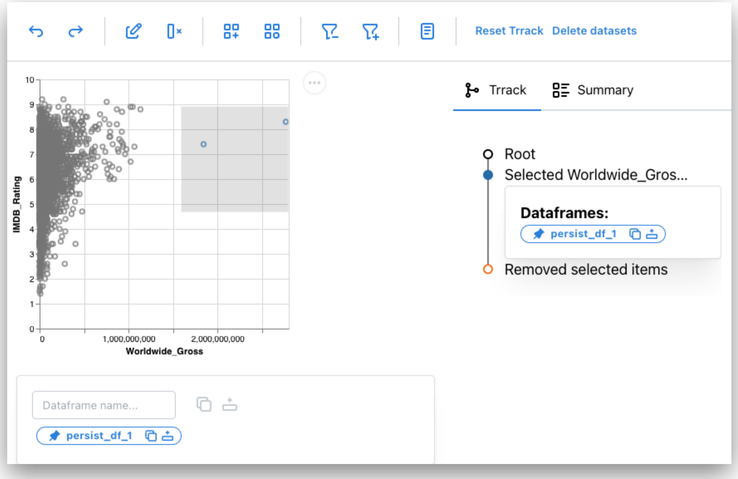
Persist works with **most Vega-Altair** charts

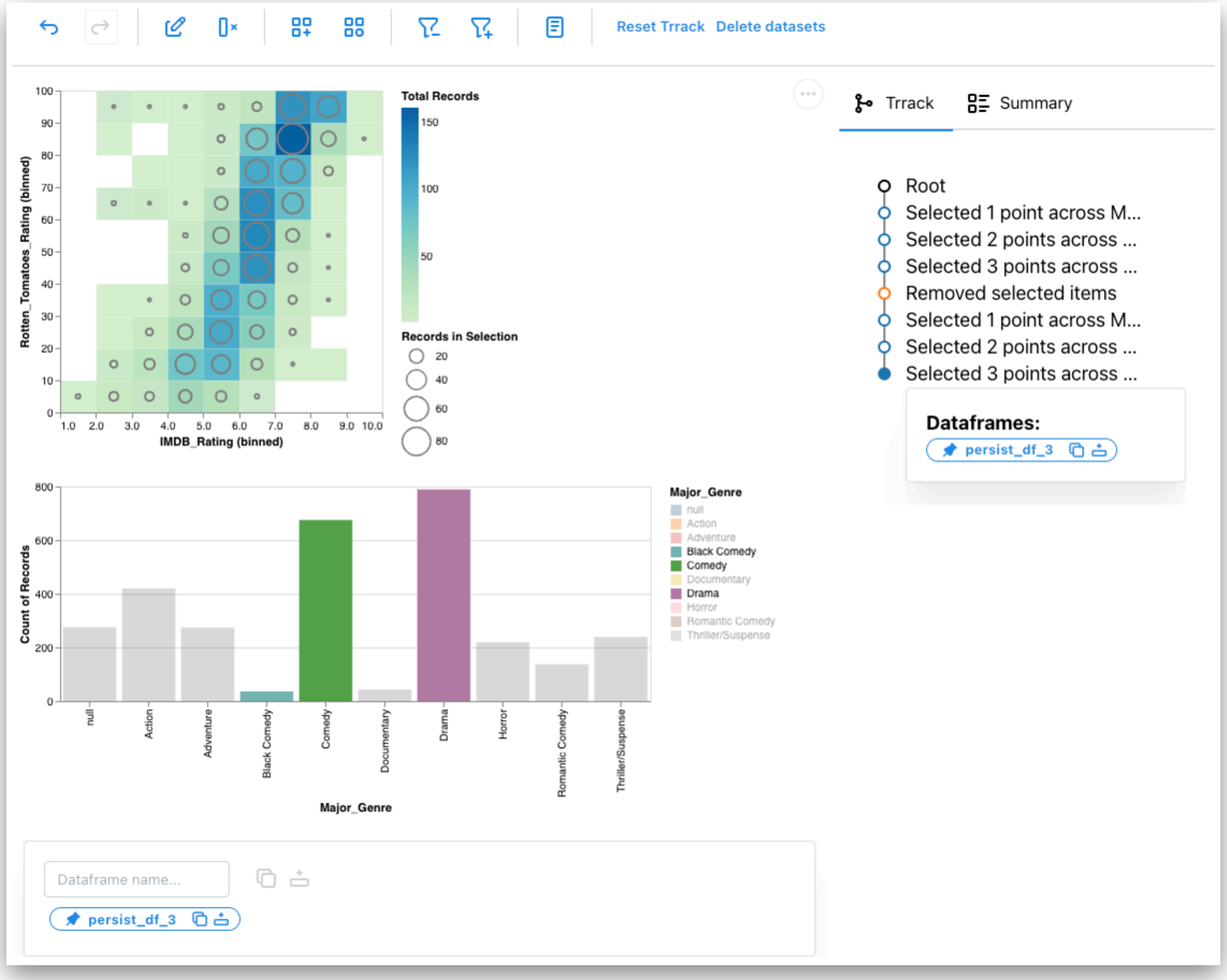
"**Listens**" to native operations (selections)

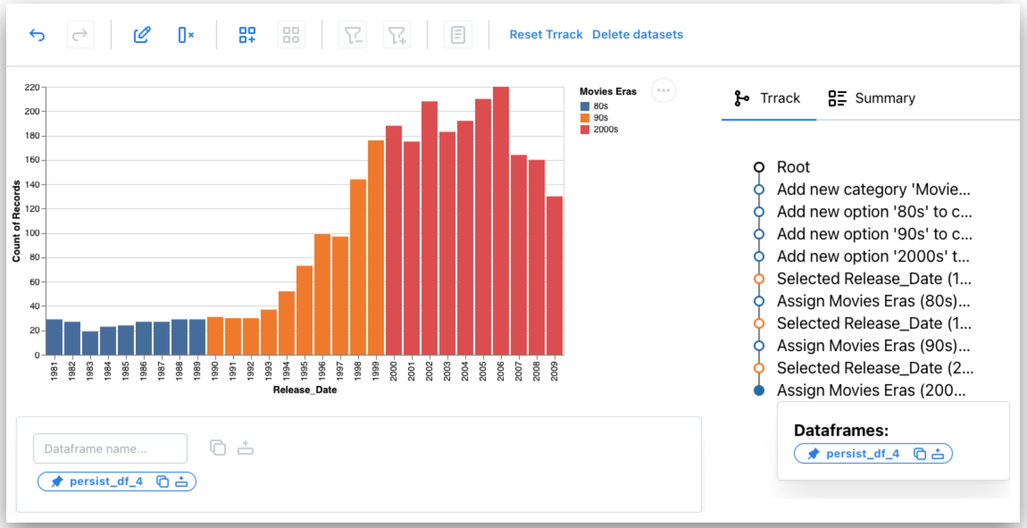
Updates Vega charts:

- Use original chart spec when possible (e.g., filters)
- Update spec when necessary (categories, labels)

EXAMPLE CHARTS







EVALUATION: IN-LAB STUDY

STUDY DESIGN

WE RECRUITED **ELEVEN** PARTICIPANTS FOR THE STUDY. PARTICIPANTS ALL HAD PRIOR EXPERIENCE WITH PYTHON AND PANDAS.

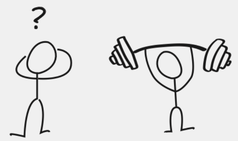
FULL FACTORIAL DESIGN

2 DATASETS ✕ 2 CONDITIONS ✕

THE ORDER OF CONDITIONS WAS **RANDOMLY** ASSIGNED.

FOR EACH CONDITION, DATASETS WERE **RANDOMLY** ASSIGNED. PARTICIPANTS NEVER SAW THE SAME DATASET TWICE

STUDY METRICS



SUBJECTIVE PERFORMANCE

TIME

ERROR

REPRODUCIBILITY

CONDITIONS

PANDAS CODING

```
[1]  IMPORT PANDAS AS PD
     DF = PD.READ_CSV(...)

[2]  DF = DF.DROP("AGE")

     DF = DF.RENAME(
       COLUMNS={'JOB': 'PROFESSION'}
     )

     DF.LOC[1, 'JOB'] = 'PRINCIPAL'
```

PERSIST EXTENSION

```
[1]  IMPORT PERSIST_EXT AS PR
     IMPORT PANDAS AS PD
     DF = PD.READ_CSV(...)
     PR.PLOT.SCATTERPLOT(...)
```

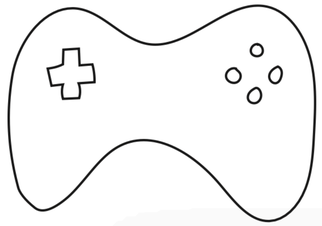
SELECT 2 VALUES

REMOVE 2 VALUES

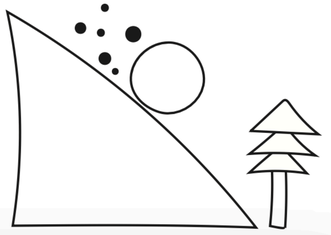
SELECT 2 VALUES

DATASETS

VIDEO GAMES



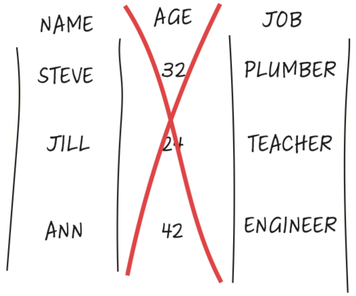
AVALANCHES



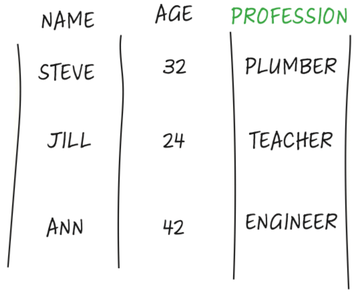
IN-LAB STUDY

TASKS

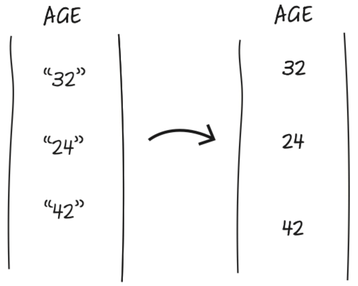
PARTICIPANTS MADE THE FOLLOWING CHANGES TO A DATASET



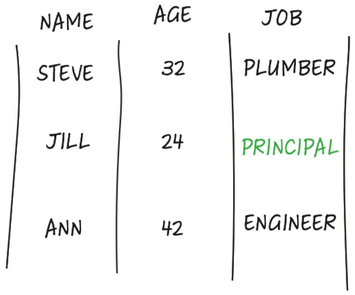
REMOVE COLUMNS



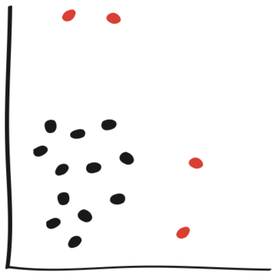
RENAME COLUMNS



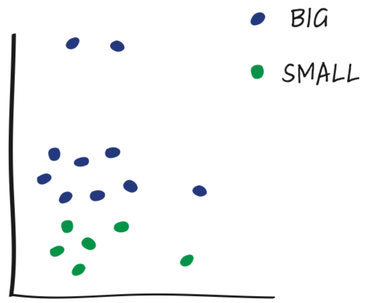
CHANGE DATA TYPE



EDIT VALUES



FILTER DATA



ADD CATEGORICAL

RESULTS

3x

times faster
with Persist

97%

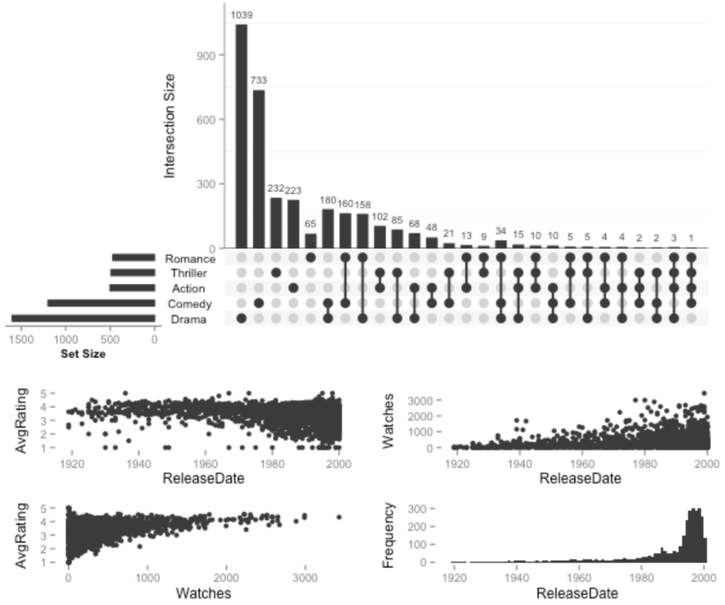
tasks **correctly** using Persist,
compared to 85% for Pandas

11/11

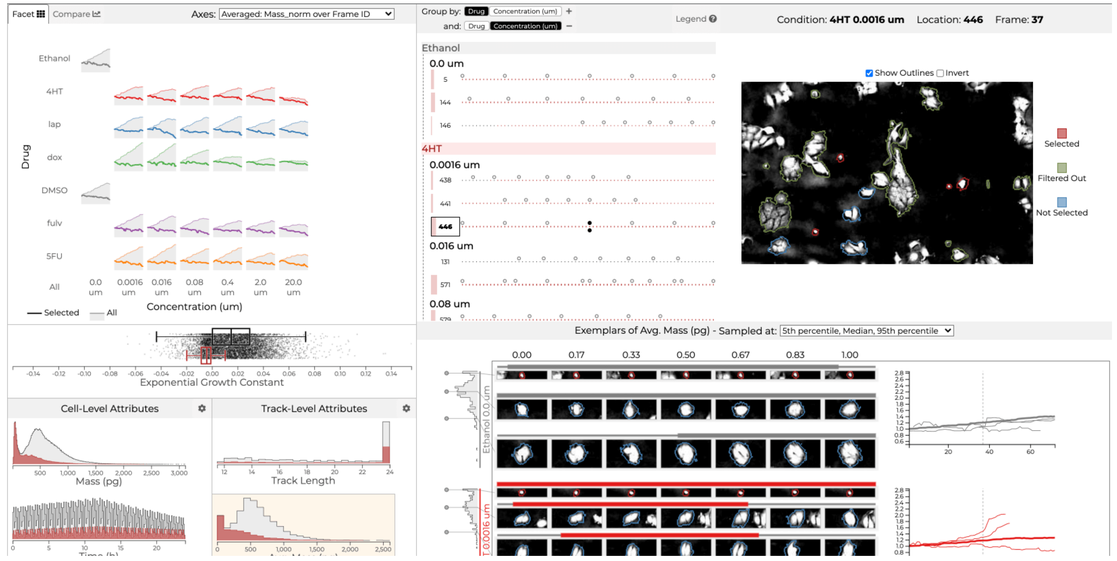
notebooks using Persist
were **reproducible**

only 7/11 using pandas
were

THE VISUALIZATION SPECTRUM



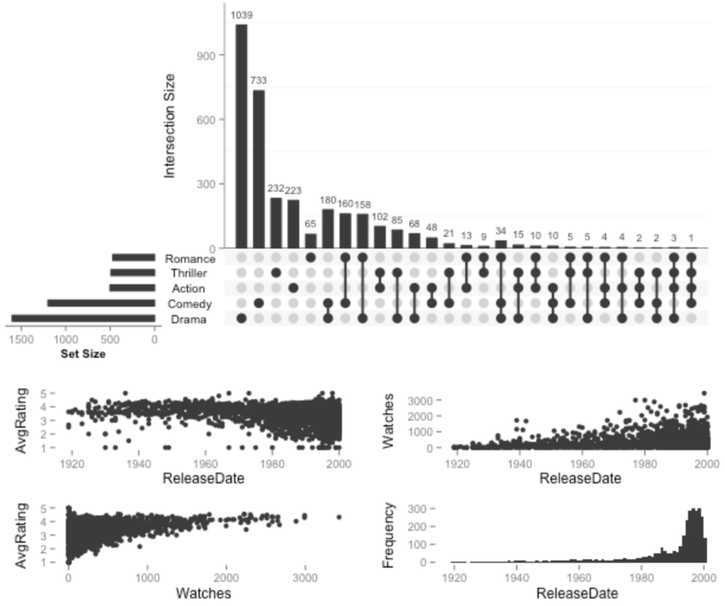

```
upset(movies,attribute.plots=list(gridrows=60,plots=list(list(plot=scatter_plot, x="ReleaseDat
list(plot=scatter_plot, x="ReleaseDate", y="Watches"),list(plot=scatter_plot, x="Watches", y="
list(plot=histogram, x="ReleaseDate")), ncols = 2))
```



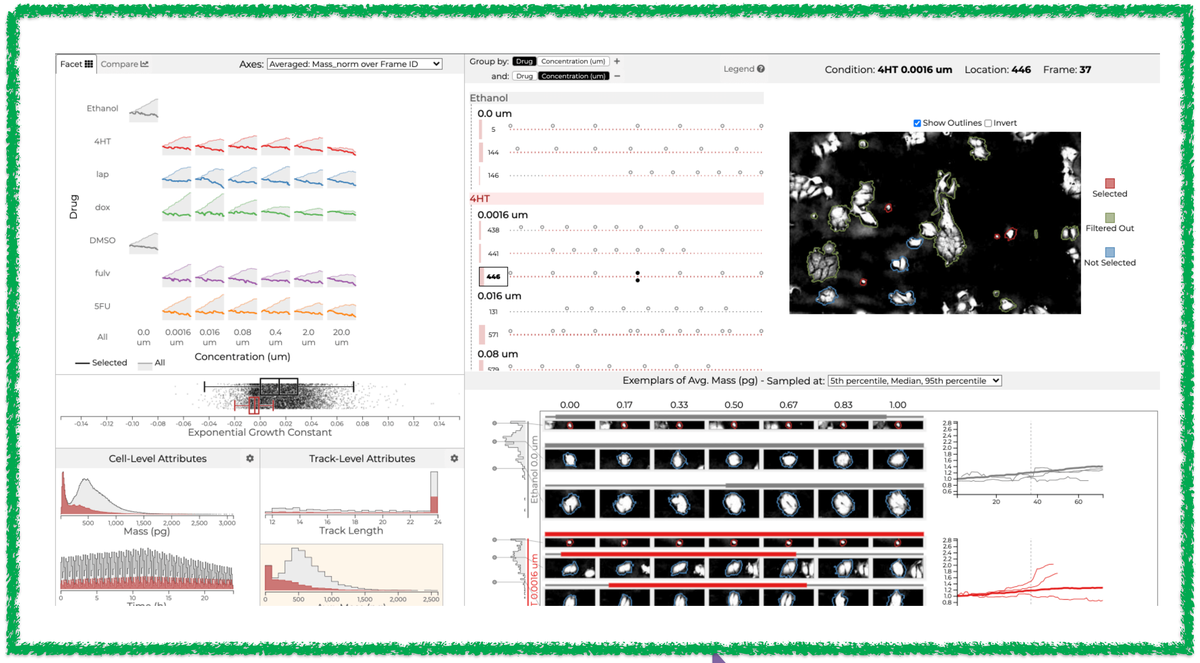
Code-generated charts

Bespoke interactive visualization systems

THE VISUALIZATION SPECTRUM




```
upset(movies,attribute.plots=list(gridrows=60,plots=list(list(plot=scatter_plot, x="ReleaseDat
list(plot=scatter_plot, x="ReleaseDate", y="Watches"),list(plot=scatter_plot, x="Watches", y="
list(plot=histogram, x="ReleaseDate")), ncols = 2))
```



Code-generated charts

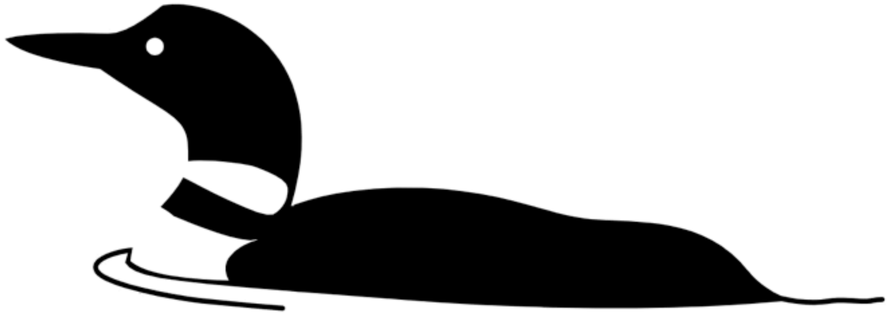
Bespoke interactive visualization systems

Loon

Using Exemplars to Visualize Large Scale Microscopy Data

Devin Lange, Eddie Polanco, Robert Judson-Torres, Thomas Zangle, Alexander Lex

http://loon.sci.utah.edu/



THE UNIVERSITY OF UTAH

visualization design lab

Zangle Lab
Department of Chemical Engineering
THE UNIVERSITY OF UTAH

UNIVERSITY OF UTAH
HUNTSMAN CANCER INSTITUTE

HEALTH
UNIVERSITY OF UTAH

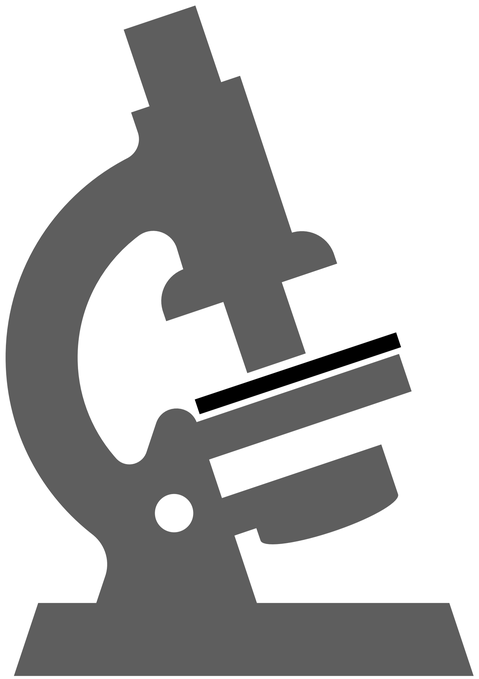
MOTIVATION: IMPROVE CANCER TREATMENT

Challenges:

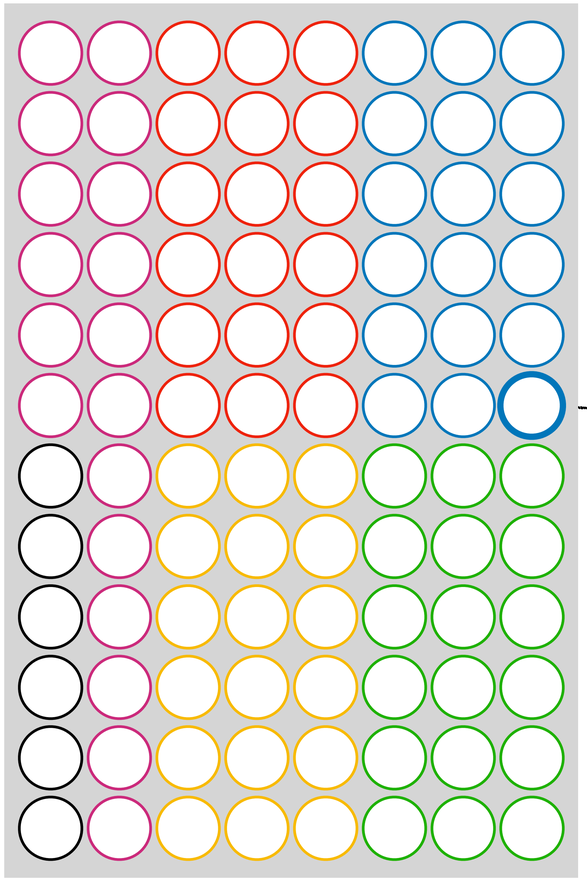
- **Cancer is heterogeneous**
- **Many treatment options exist**

Goal:

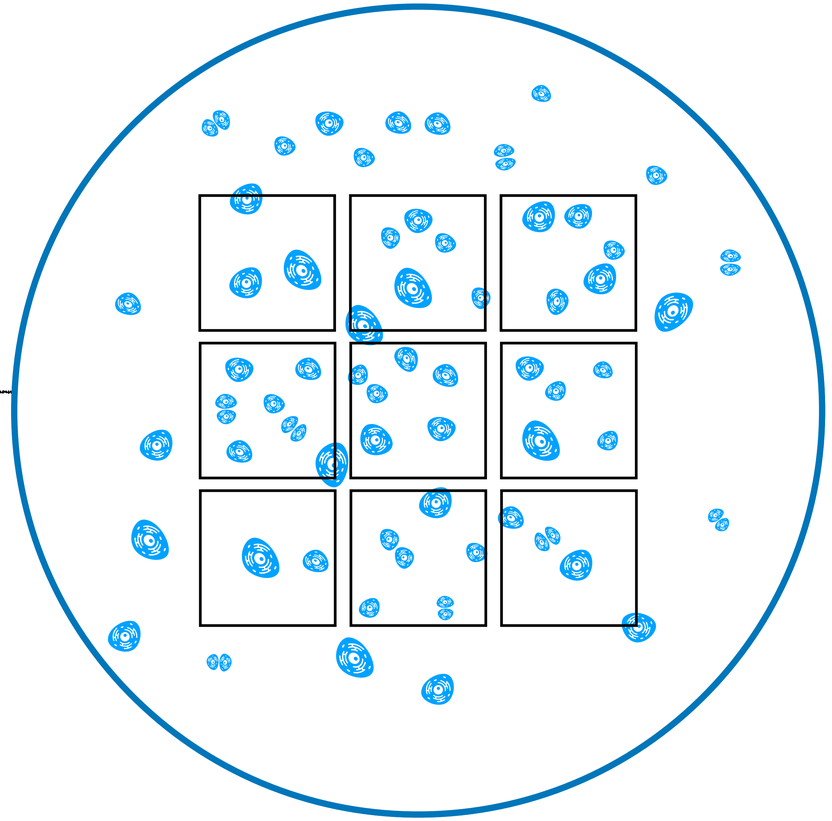
- **Identify effective treatment option for individuals through time-series microscopy**



Quantitative Phase Microscope

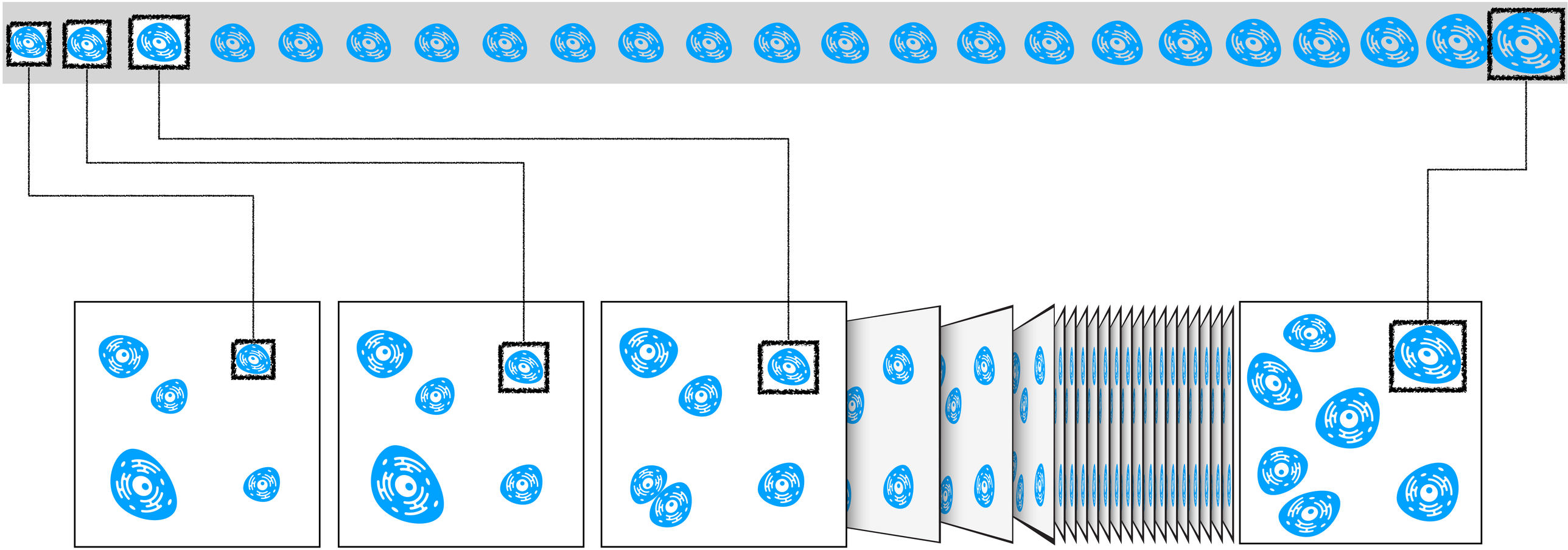


96-Well Plate

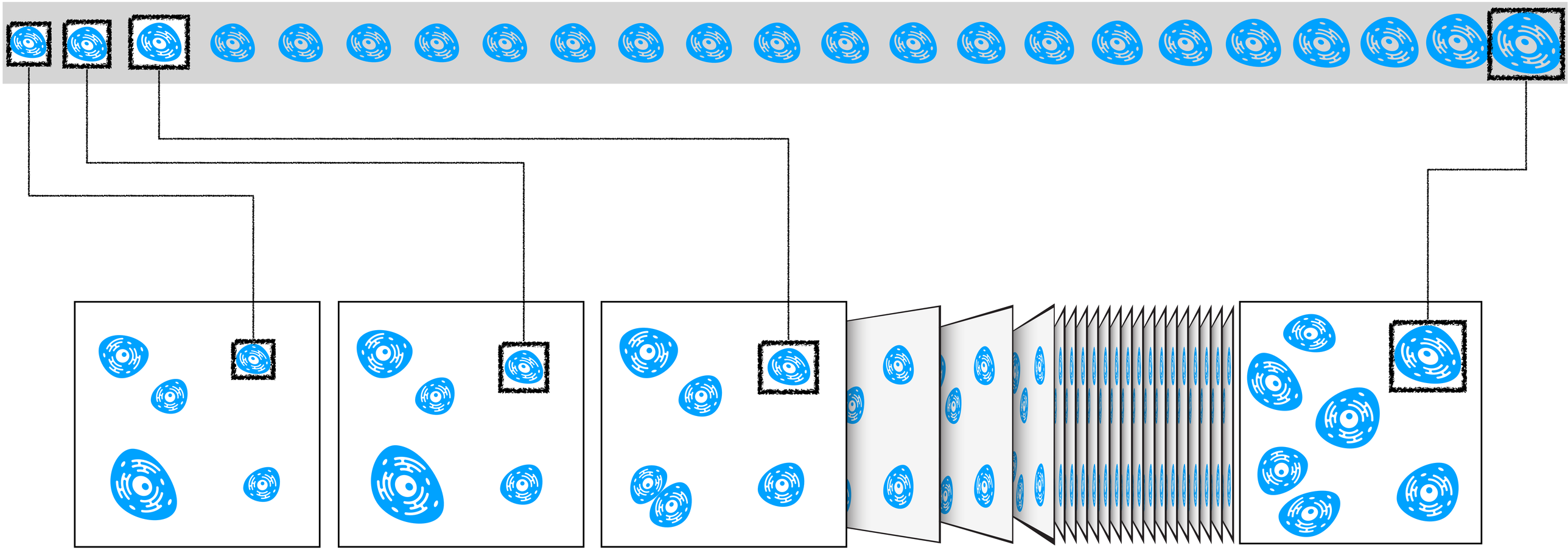


Imaging a Single Well

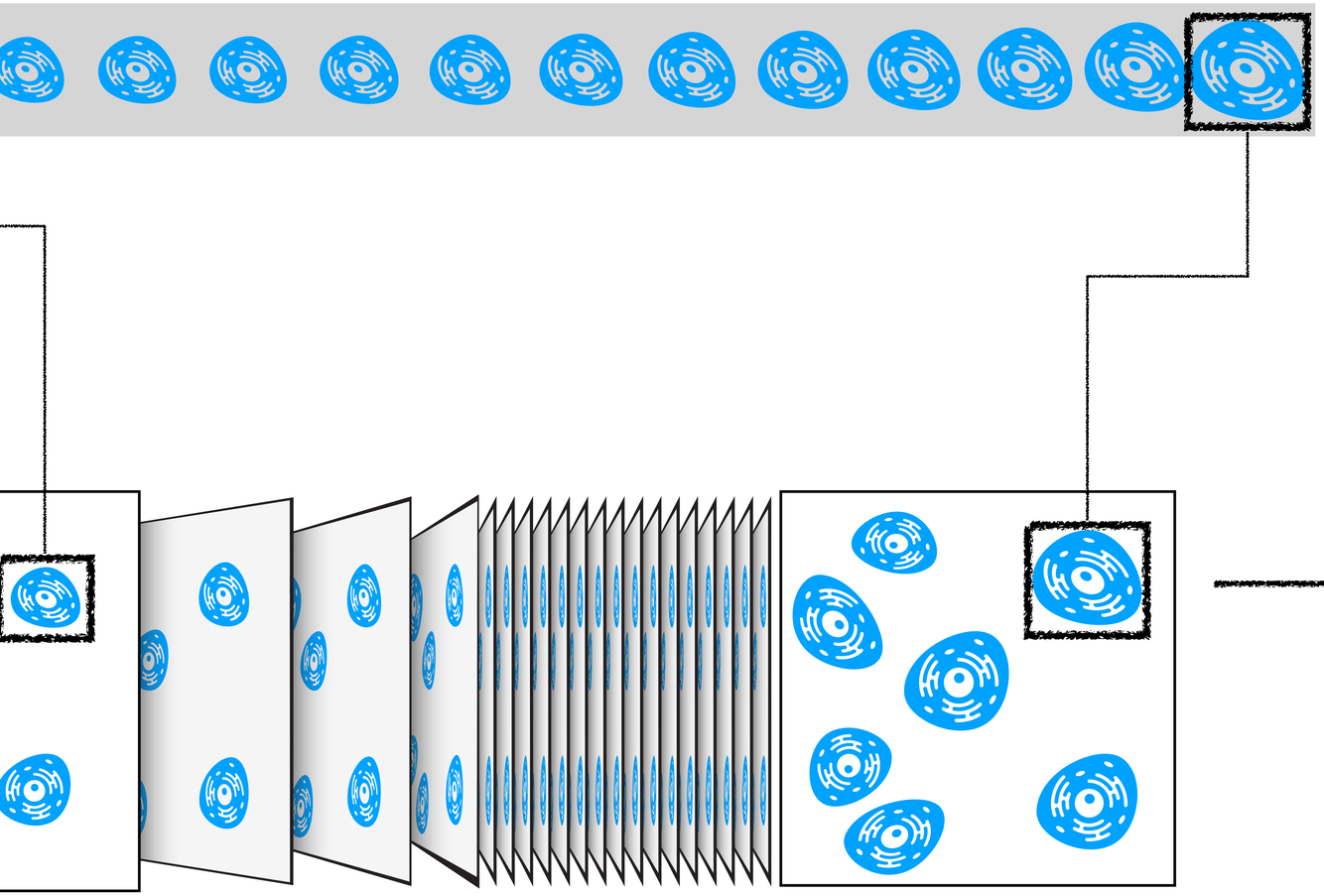
Tracking Cell Over Time



Tracking Cell Over Time



Cell and Track Attributes

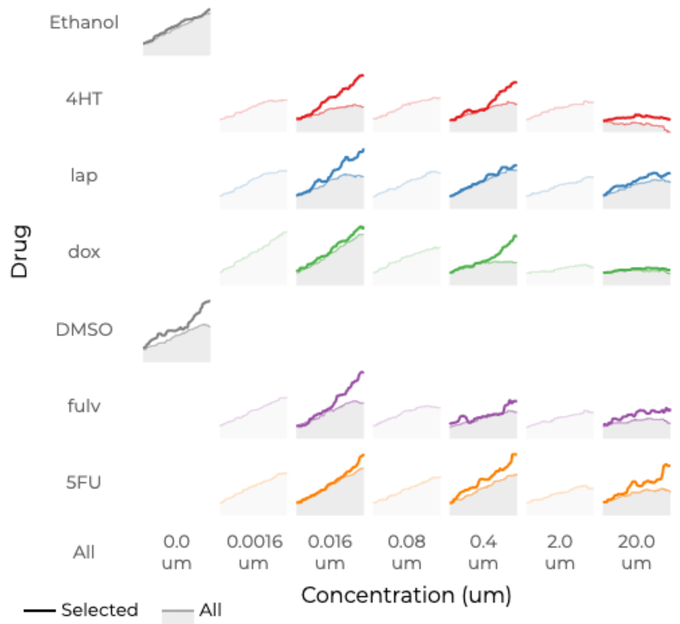


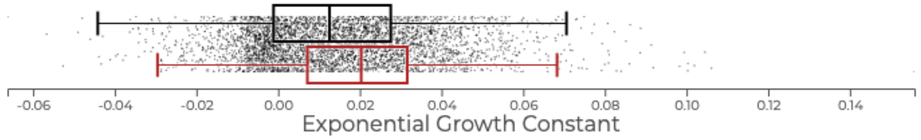
Average Mass
Growth Rate
Track Length

Current Mass
Shape Factor
Density

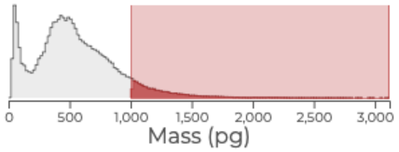
Facet | Compare

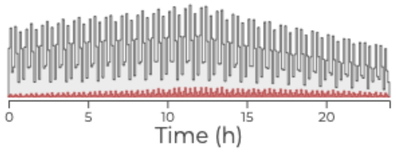
Axes: Averaged: Mass_norm over Frame ID

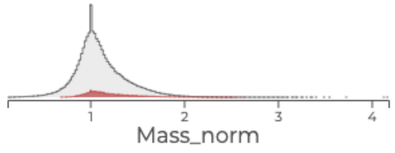




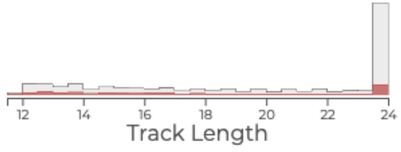
Cell-Level Attributes

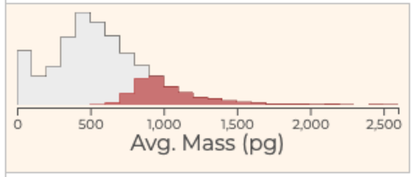


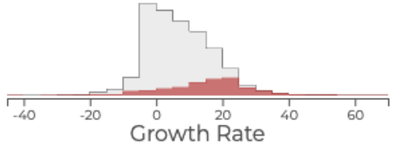




Track-Level Attributes





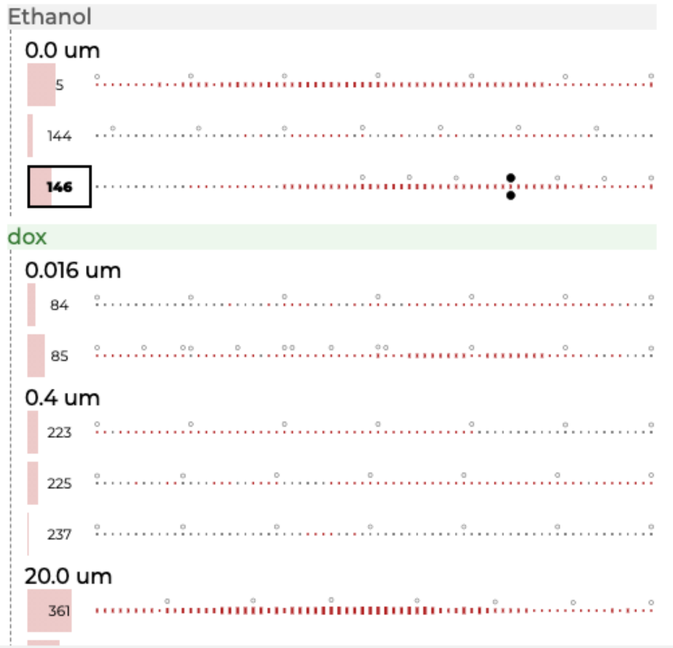


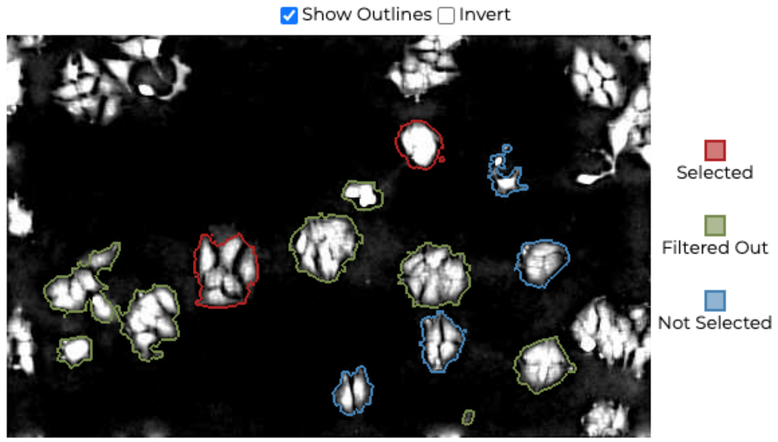
Group by: Drug | Concentration (um) +

and: Drug | Concentration (um) −

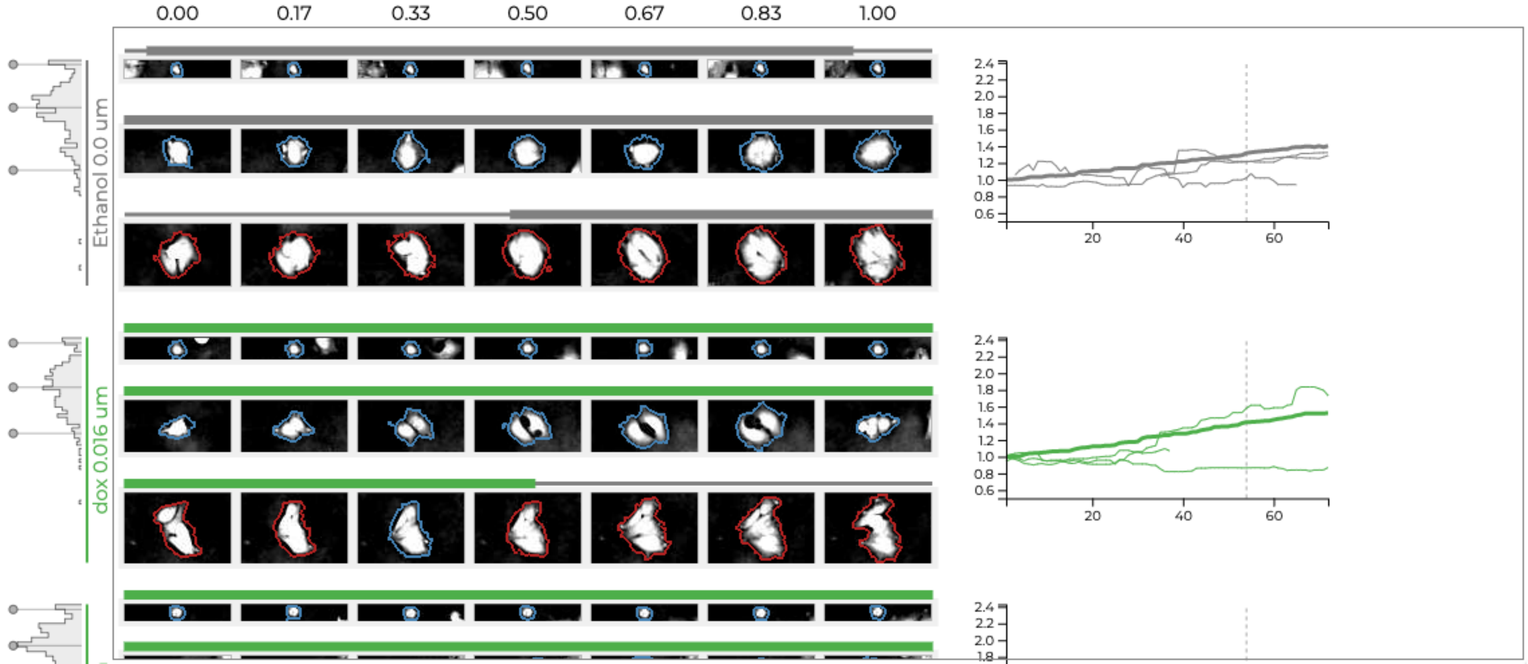
Legend

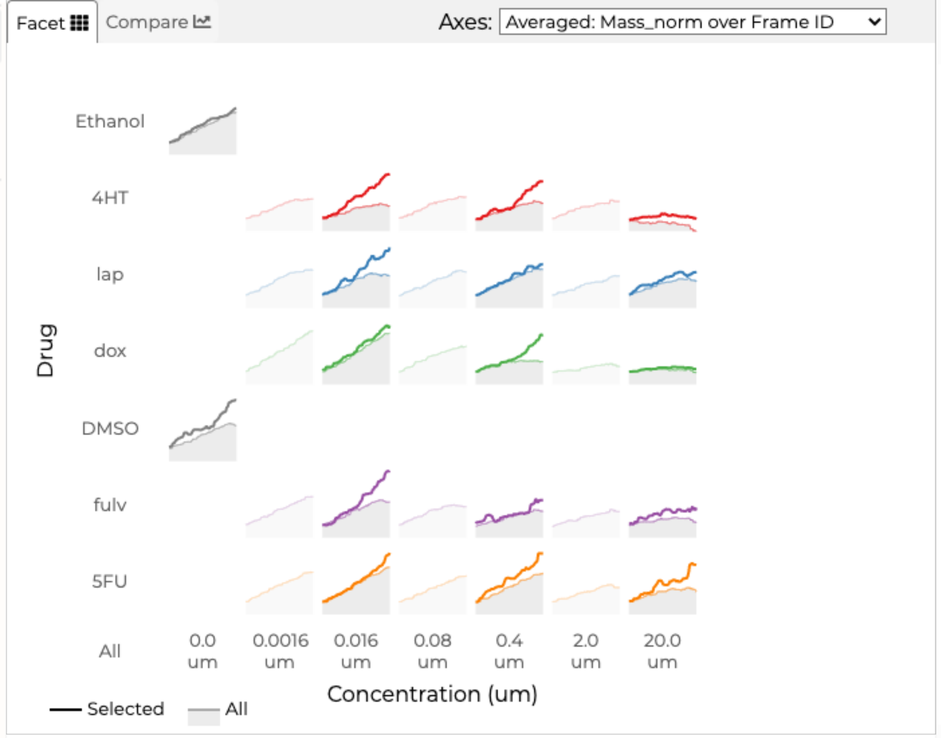
Condition: **Ethanol 0.0 um** Location: **146** Frame: **54**



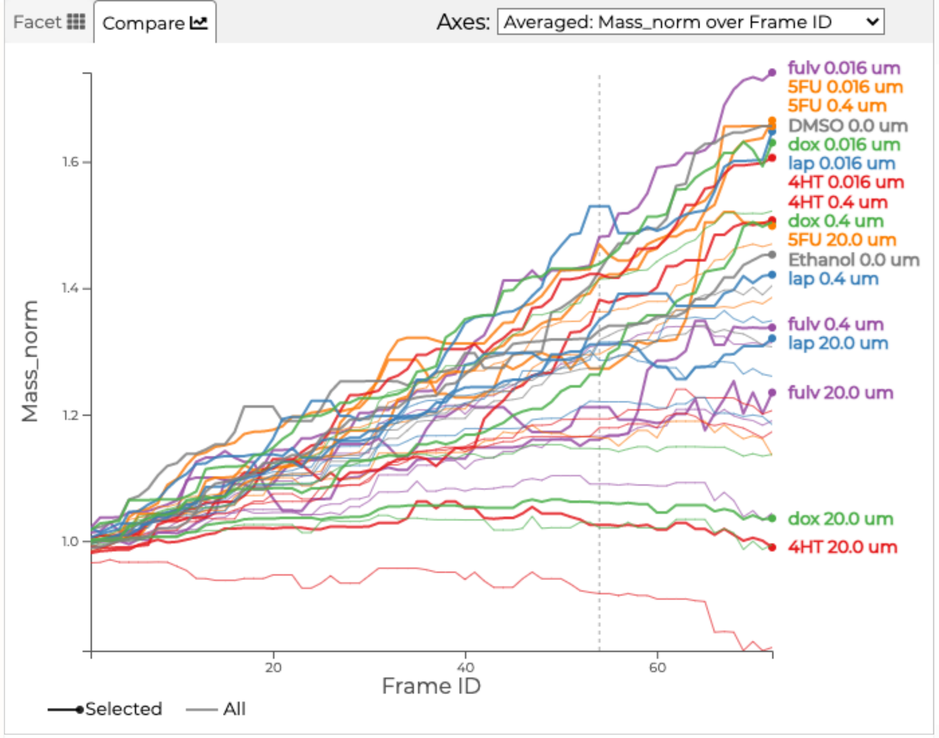


Exemplars of Avg. Mass (pg) - Sampled at: 5th percentile, Median, 95th percentile

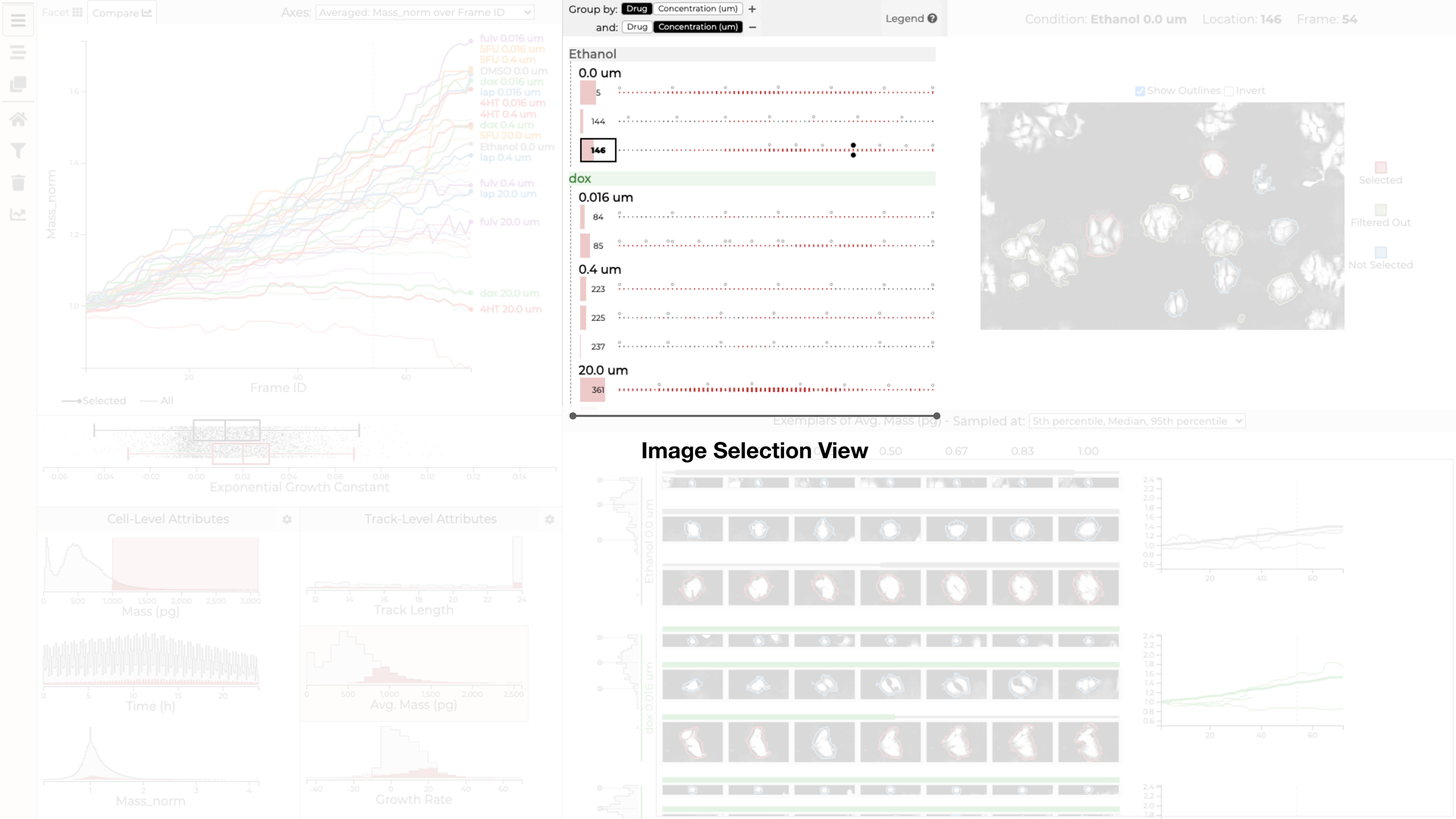




Condition Selector



Growth Curves


Facet
Compare
Axes: Averaged: Mass_norm over Frame ID
fulv 0.016 um
5FU 0.016 um
5FU 0.4 um
DMSO 0.0 um
dox 0.016 um
lap 0.016 um
4HT 0.016 um
4HT 0.4 um
dox 0.4 um
5FU 20.0 um
Ethanol 0.0 um
lap 0.4 um
fulv 0.4 um
lap 20.0 um
fulv 20.0 um
dox 20.0 um
4HT 20.0 um
Mass_norm
Frame ID
Selected
All
Exponential Growth Constant
Cell-Level Attributes
Track-Level Attributes
Mass (pg)
Track Length
Time (h)
Avg. Mass (pg)
Mass_norm
Growth Rate
Group by: Drug Concentration (um) +
and: Drug Concentration (um) −
Legend
Ethanol
0.0 um
5
144
146
dox
0.016 um
84
85
0.4 um
223
225
237
20.0 um
361
Condition: Ethanol 0.0 um
Location: 146
Frame: 54
Show Outlines
Invert
Selected
Filtered Out
Not Selected
Exemplars of Avg. Mass (pg) - Sampled at: 5th percentile, Median, 95th percentile
Image Selection View
0.50
0.67
0.83
1.00
Ethanol 0.0 um
dox 0.016 um

Facet | Compare | Axes: Averaged: Mass_norm over Frame ID

Group by: Drug | Concentration (um) + and: Drug | Concentration (um) −

Legend

Condition: **Ethanol 0.0 um** Location: **146** Frame: **54**

Ethanol
- 0.0 um: 5, 144, 146
dox
- 0.016 um: 84, 85
- 0.4 um: 223, 225, 237
- 20.0 um: 361

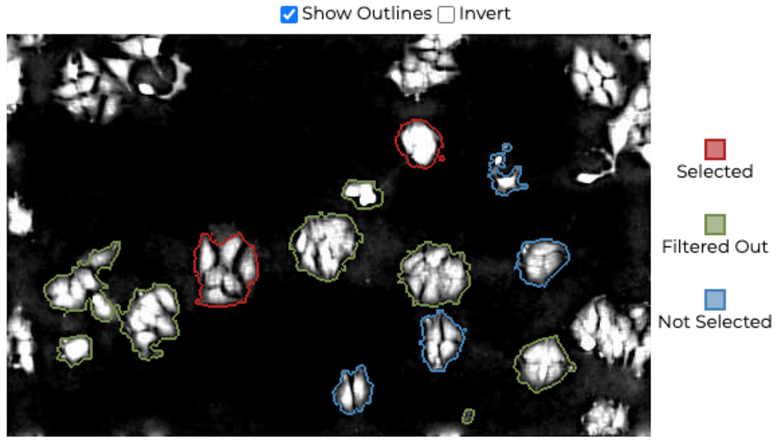
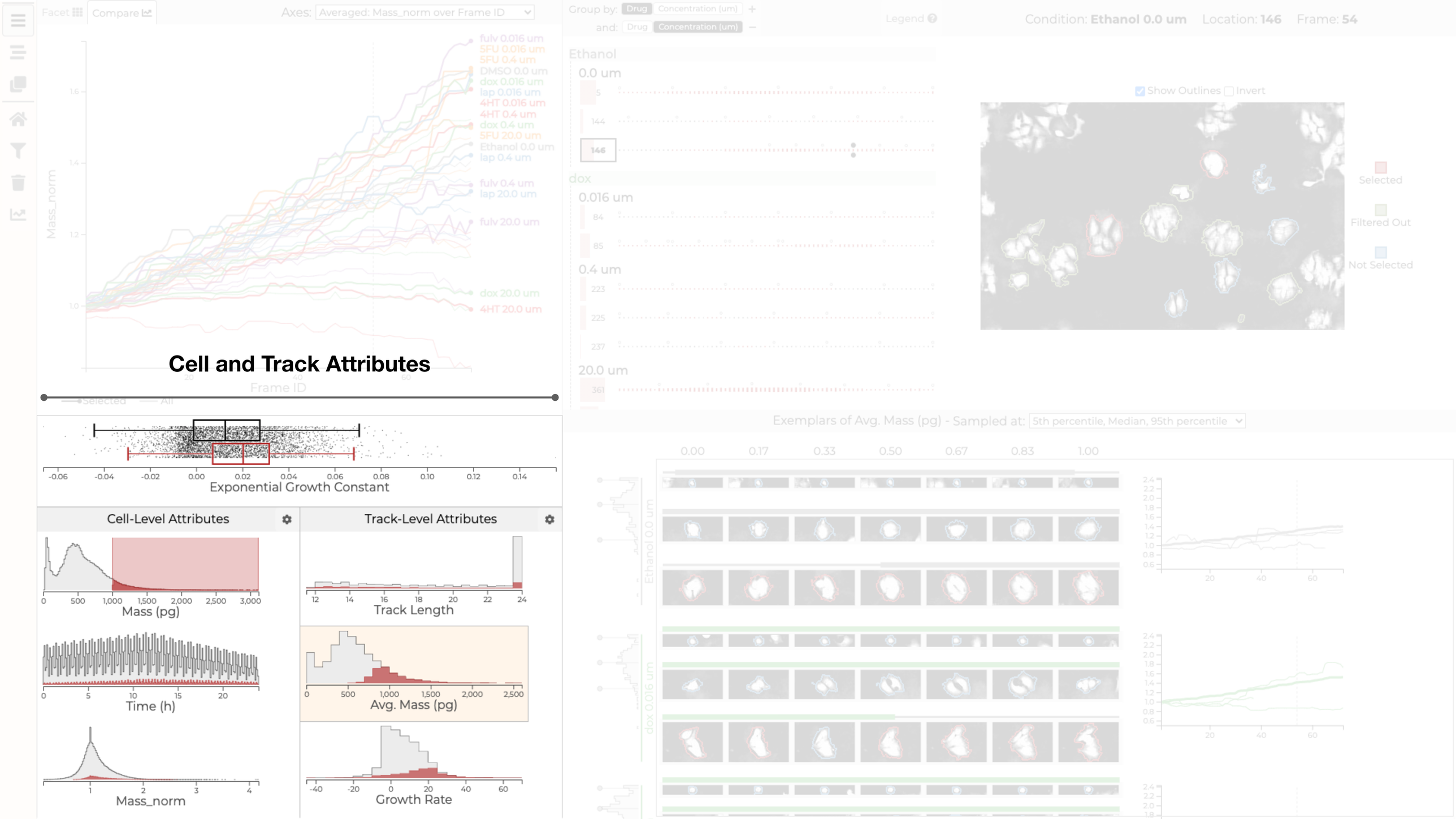


Image View

Exemplars of Avg. Mass (pg) - Sampled at: 5th percentile, Median, 95th percentile


Facet
Compare
Axes: Averaged: Mass_norm over Frame ID
fulv 0.016 um
5FU 0.016 um
5FU 0.4 um
DMSO 0.0 um
dox 0.016 um
lap 0.016 um
4HT 0.016 um
4HT 0.4 um
dox 0.4 um
5FU 20.0 um
Ethanol 0.0 um
lap 0.4 um
fulv 0.4 um
lap 20.0 um
fulv 20.0 um
dox 20.0 um
4HT 20.0 um
Mass_norm
Frame ID
Group by: Drug Concentration (um)
and: Drug Concentration (um)
Legend
Condition: Ethanol 0.0 um
Location: 146
Frame: 54
Ethanol
0.0 um
dox
0.016 um
0.4 um
20.0 um
Show Outlines
Invert
Selected
Filtered Out
Not Selected
Cell and Track Attributes
Selected
All
Exponential Growth Constant
Cell-Level Attributes
Track-Level Attributes
Mass (pg)
Track Length
Time (h)
Avg. Mass (pg)
Mass_norm
Growth Rate
Exemplars of Avg. Mass (pg) - Sampled at:
5th percentile, Median, 95th percentile
0.00
0.17
0.33
0.50
0.67
0.83
1.00
Ethanol 0.0 um
dox 0.016 um

Facet | Compare

Axes: Averaged: Mass_norm over Frame ID

Cell-Level Attributes

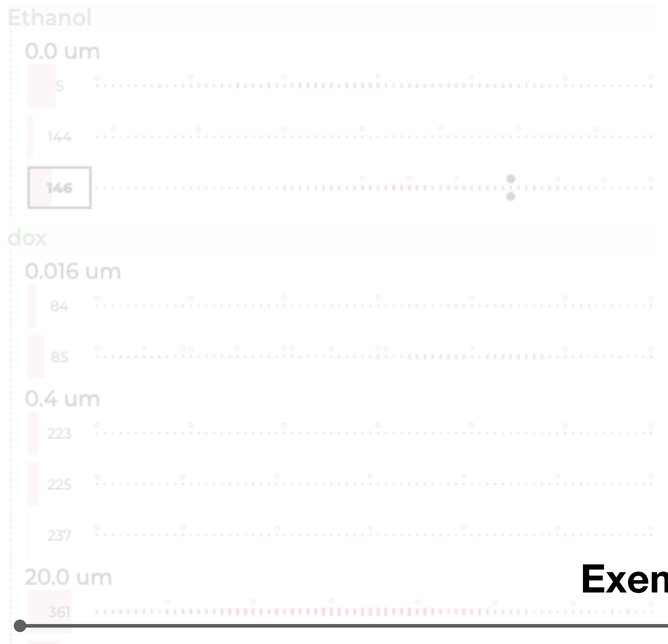
Track-Level Attributes

Group by: Drug | Concentration (um) +

and: Drug | Concentration (um) −

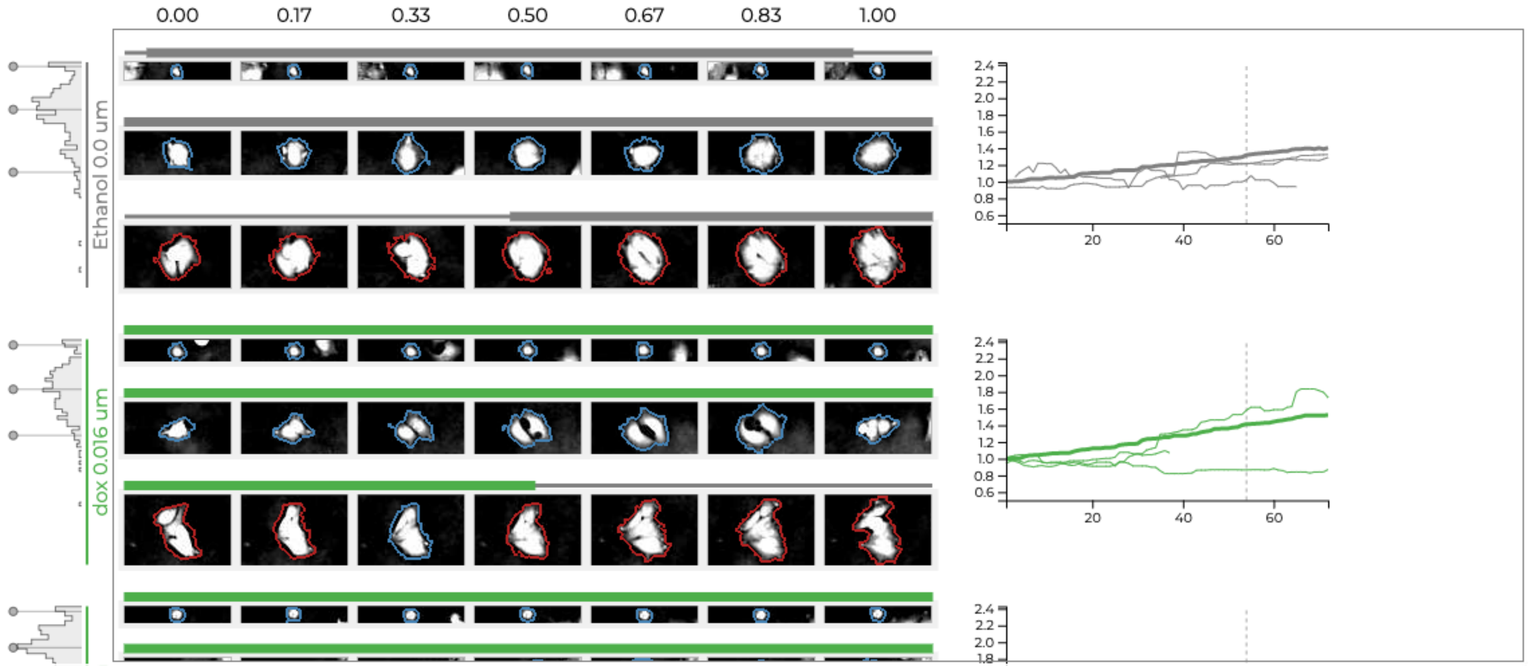
Legend

Condition: **Ethanol 0.0 um** Location: **146** Frame: **54**



Exemplar Tracks View

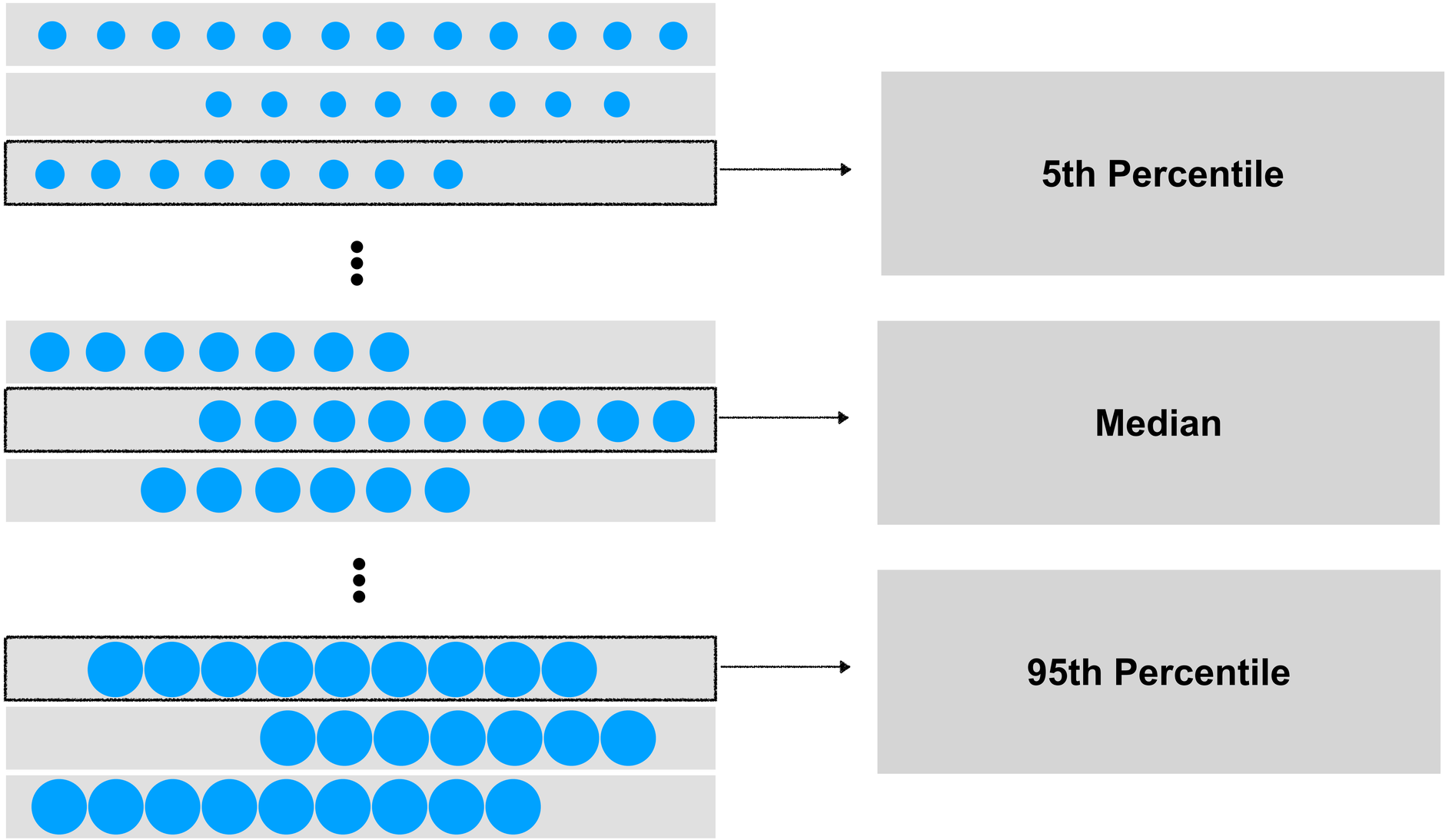
Exemplars of Avg. Mass (pg) - Sampled at: 5th percentile, Median, 95th percentile



WHY EXEMPLARS?

Want to see good representative examples.

Tricky to do given the scale of the data.

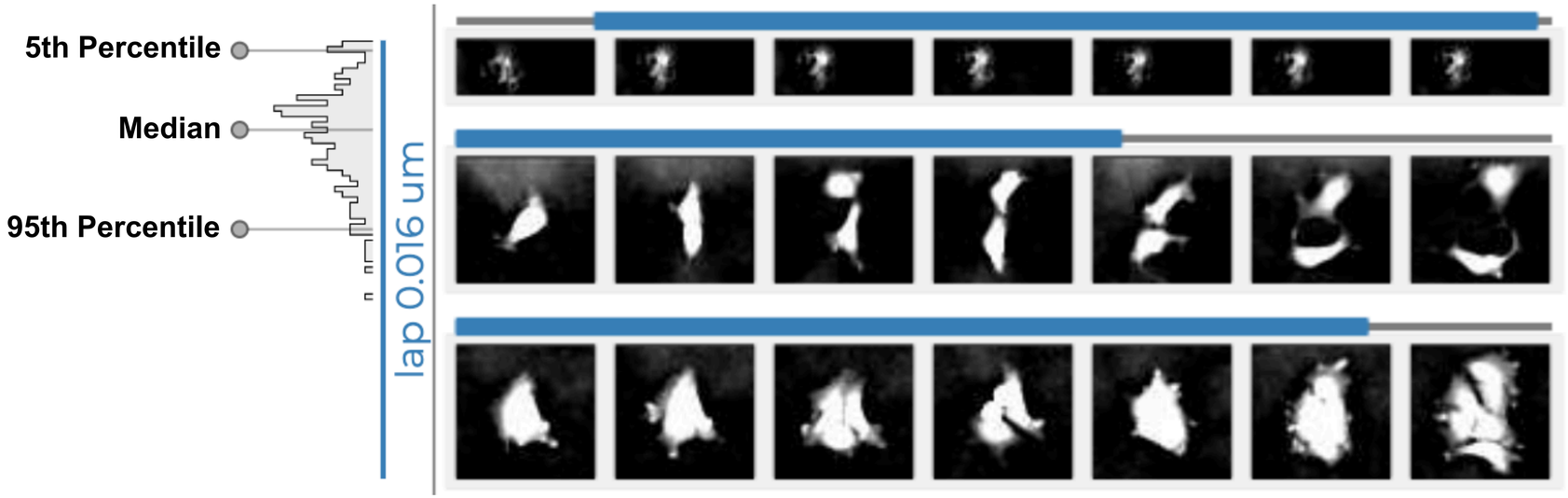


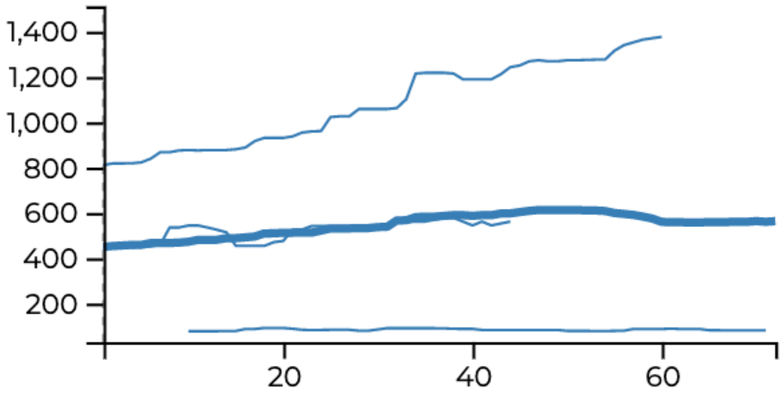
Sort by

Mass
Shape Factor
Growth Rate

Sample Tracks

5th, Median, 95th
Min, Median, Max
25th, Median, 75th





Final Display

loon.sci.utah.edu/detailedView/Example_MCF7

Facet | Compare

Axes: Averaged: Mass_norm over Frame ID

Drug: Ethanol, 4HT, lap, dox, DMSO, fulv, 5FU, All

Concentration (um): 0.0 um, 0.0016 um, 0.016 um, 0.08 um, 0.4 um, 2.0 um, 20.0 um

Selected | All

Exponential Growth Constant

Cell-Level Attributes

Mass (pg)

Time (h)

Track-Level Attributes

Track Length

Avg. Mass (pg)

Group by: Drug, Concentration (um)

and: Drug, Concentration (um)

Legend

Ethanol

0.0 um

5

144

146

dox

0.016 um

84

85

0.4 um

223

225

237

Condition: **Ethanol 0.0 um** Location: **144** Frame: **3**

Show Outlines | Invert

Selected

Filtered Out

Not Selected

Exemplars of Avg. Mass (pg) - Sampled at: 5th percentile, Median, 95th percentile

0.00 0.17 0.33 0.50 0.67 0.83 1.00

Ethanol 0.0 um

dox 0.016 um

https://loon.sci.utah.edu/

When to use exemplars?

Rich contextual data that cannot be summarized

Dataset is sufficiently large

SO, WHAT'S BETTER?

WHAT SPEAKS FOR BESPOKE SYSTEMS

Do you have **unique and complex data** (not a table)?

Is your analysis a **recurring workflow**?

Do you have **a lot of data** that you need to “review”?

Do you **need more** than what’s in matplotlib or ggplot?

Probably go with a bespoke system

Next decision: buy vs build

Alexander Lex
@alexander_lex
http://alexander-lex.net

Thanks to: Max Lisnic, Devin Lange, Kiran Gadhave, Zach Cutler, Jack Wilburn, Ishrat Jahan Eliza, Jake Wagoner, Haihan Lin, Jen Rogers, Carolina Nobre, Derya Akbaba, Marc Streit, Marina Kogan, Lane Harrison, Jochen Görtler, Oliver Deussen, Miriah Meyer, Jeff Phillips, Samuel Gratzl, Holger Stitz, Nils Gehlenborg, Hendrik Strobelt, Romain Vuillemot, Hanspeter Pfister, and many others!

visualization
design lab