Diving into JUMP Cell Painting data: Facilitating novel discoveries through accessible tools

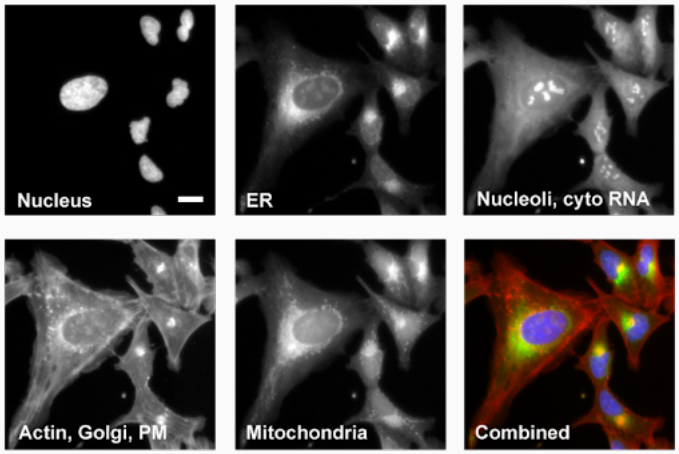
Alán F. Muñoz

2024/05/06

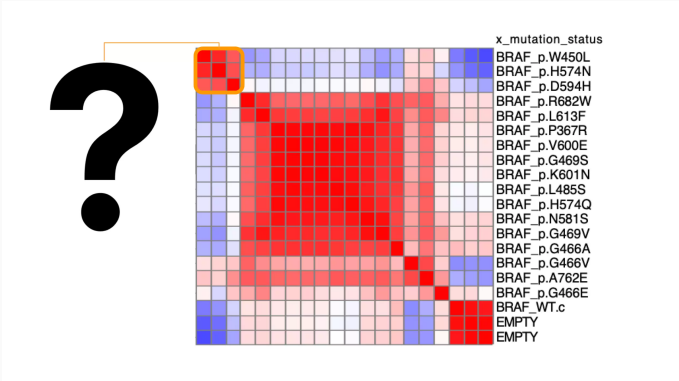
Introduction: Turns out high content imaging is hard.

The JUMP-CP Consortium produced large diverse datasets

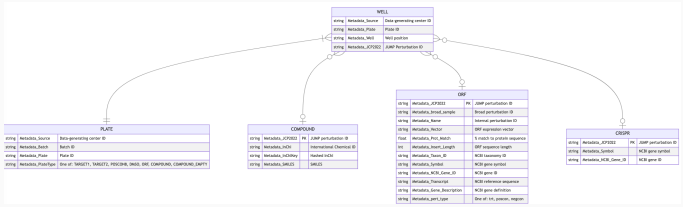
- Raw images



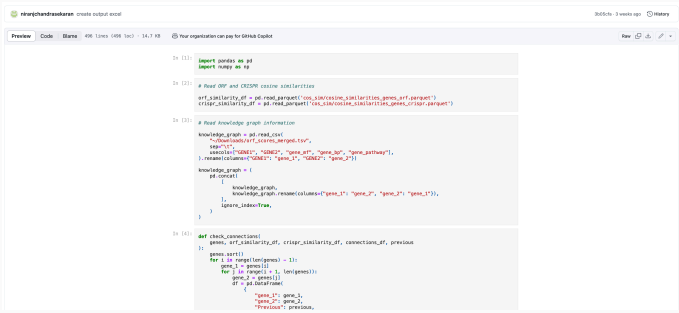
- Morphological profiles



- Metadata tables



- Notebooks



Why is this a problem of scale?

- Loads of data
  - ~115k chemical perturbations
  - ~15k genes (KO'd, overexpressed or both).

Why is this a problem of scale?

- Scattered metadata
  - How do we visualise the images from which a profile was produced?
  - Are these two perturbations from the same batch/plate?

Why is this a problem of scale?

- Diverse backgrounds
  - How can biologists with limited coding experience benefit?
  - How can developers build their own tools?

We aim to help scientists further their research

We use our own use-cases to make the data useful

I have -omics data for X modified gene or chemical compound.

- How do cells affected by X look?

We use our own use-cases to make the data useful

I have -omics data for X modified gene or chemical compound.

- What else produces similar morphologies?

We use our own use-cases to make the data useful

I have -omics data for X modified gene or chemical compound.

- What are the distinctive features of X-perturbed cells?

We use our own use-cases to make the data useful

I have -omics data for X modified gene or chemical compound.

- Can my X be found under a different name?

Ease of access: We preprocess data publish it using WebAssembly

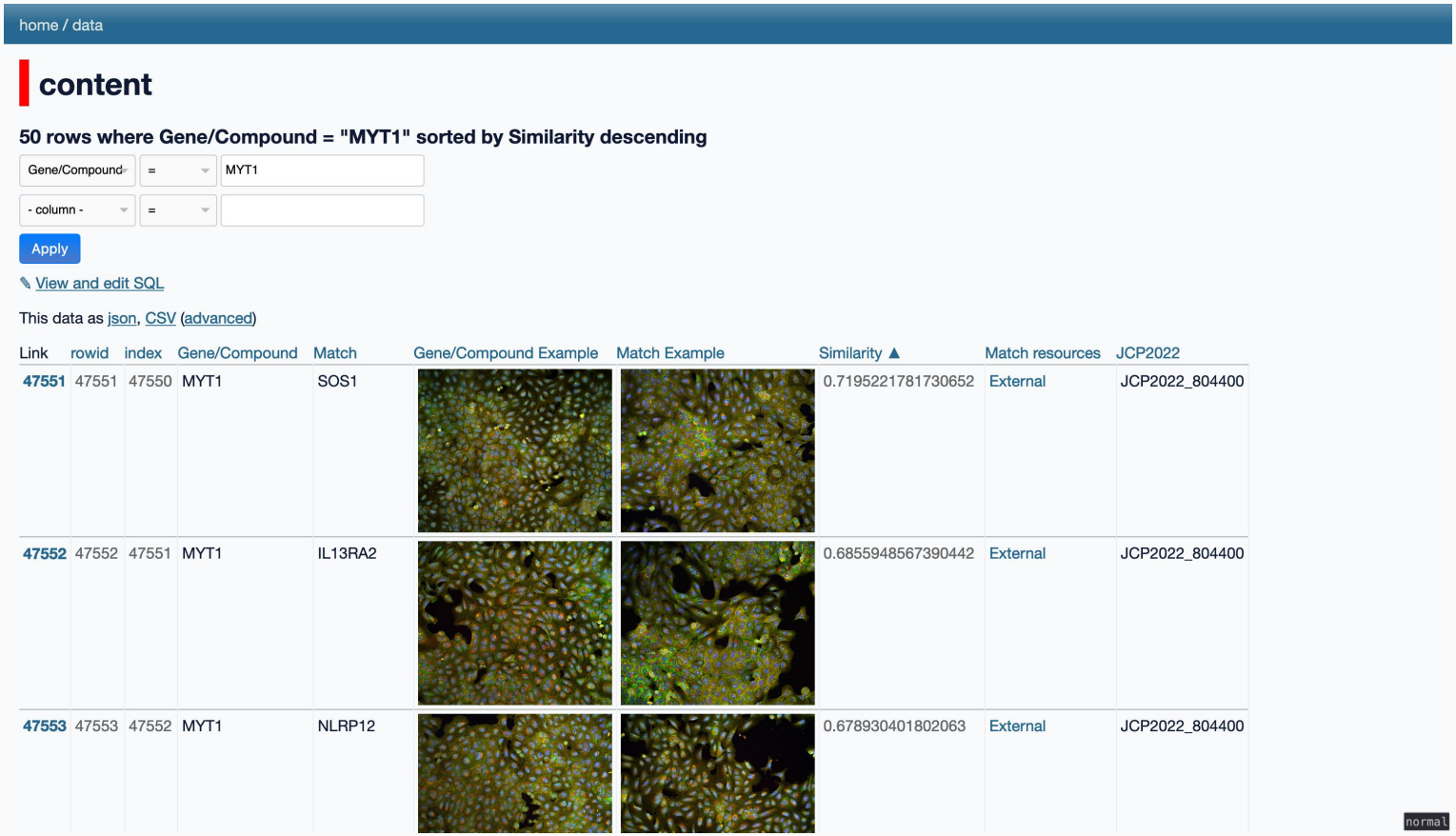
- Process data: **`jump_rr`** (JUMP Round-Robin) is our tool to perform pairwise computations on profiles and their metadata on GPUs.

Ease of access: We preprocess data publish it using WebAssembly

- Publish results: **`Datasette`** is a Python libary to visualise, query and edit databases.

What else produces similar morphologies?

`matches`: Find the most similar (and dissimilar) perturbations.



home / data

content

50 rows where Gene/Compound = "MYT1" sorted by Similarity descending

Gene/Compound = MYT1

- column - =

Apply

View and edit SQL

This data as json, CSV (advanced)

Link	rowid	index	Gene/Compound	Match	Gene/Compound Example	Match Example	Similarity ▲	Match resources	JCP2022
47551	47551	47550	MYT1	SOS1			0.7195221781730652	External	JCP2022_804400
47552	47552	47551	MYT1	IL13RA2			0.6855948567390442	External	JCP2022_804400
47553	47553	47552	MYT1	NLRP12			0.678930401802063	External	JCP2022_804400

normal

What else produces similar morphologies?

What are the distinctive features of X-perturbed cells?

`feature`: Find the most distinctive features of a perturbation, or amongst all.

67,700 rows

- column - | = | Apply

✎ View and edit SQL

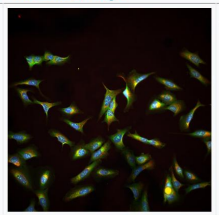
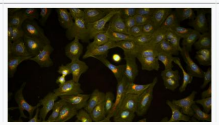
This data as json, CSV (advanced)

Channel 11 ✖

- RNA 12,350
- Mito 12,050
- ER 11,750
- DNA 11,450
- AGP 11,150
- - 5,100
- OrigDNA 800
- OrigRNA 800
- OrigAGP 750
- OrigER 750
- OrigMito 750

Mask 4 ✖

- Nuclei 19,100
- Cells 17,500
- Cytoplasm 17,050
- Image 14,050

Link	rowid ▼	index	Mask	Feature	Channel	Statistic	Gene/Compound	Metadata_image	Median	JCP2022
1	1	0	Cells	Correlation_K_Mito	AGP	8.349928508374835e-29	AVPR2		-0.21309228714749423	JCP2022_905491
2	2	1	Cells	Correlation_K_Mito	AGP	3.325815194945794e-19	EEF1AKMT2		0.13034929387328936	JCP2022_912740

normal

What are the distinctive features of X-perturbed cells?

How do cells affected by X look?

`gallery`: Explore all the images with an associated profile.

12 rows where Gene/Compound in ["MYT1", "PROZ"] sorted by rowid

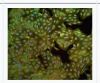
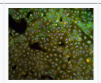
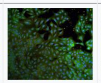
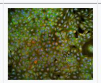
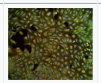
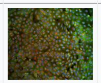
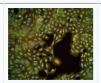
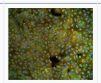
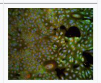
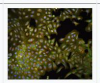
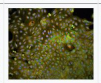
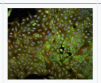
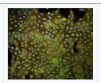
Gene/Compound | in | MYT1, PROZ

- column - | =

Apply

View and edit SQL

This data as json, CSV (advanced)

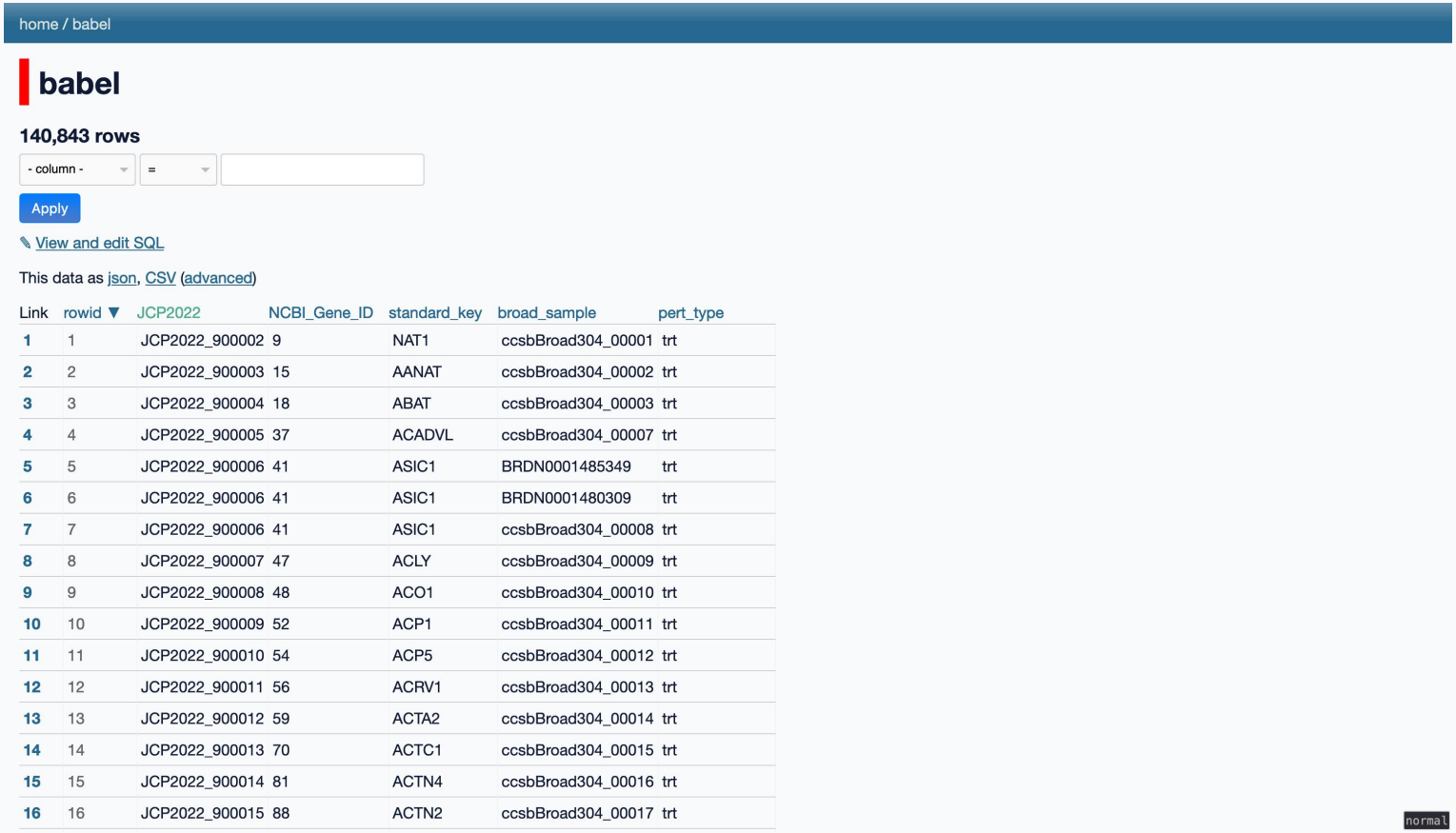
Link	rowid ▼	index	Gene/Compound	External resources	Metadata_JCP2022	Foci 0	Foci 1	Foci 2	Foci 3	Foci 4	Foci 5	Foci 6	Foci 7	Foci 8
6	6	5	PROZ	External	JCP2022_805564									
1055	1055	1054	MYT1	External	JCP2022_804400									
9671	9671	9670	PROZ	External	JCP2022_805564									
10704	10704	10703	MYT1	External	JCP2022_804400									
19344	19344	19343	PROZ	External	JCP2022_805564									
20390	20390	20389	MYT1	External	JCP2022_804400									
29042	29042	29041	MYT1	External	JCP2022_804400									
29521	29521	29520	PROZ	External	JCP2022_805564									

normal

How do cells affected by X look?

Does X have a different name in other databases?

`broad_babel`: Translates IDs (JUMP, Symbol/InChiKey, Entrez id, Broad) and which are controls.



home / babel

babel

140,843 rows

- column - | = | Apply

View and edit SQL

This data as json, CSV (advanced)

Link	rowid ▼	JCP2022	NCBI_Gene_ID	standard_key	broad_sample	pert_type
1	1	JCP2022_900002	9	NAT1	ccsbBroad304_00001	trt
2	2	JCP2022_900003	15	AANAT	ccsbBroad304_00002	trt
3	3	JCP2022_900004	18	ABAT	ccsbBroad304_00003	trt
4	4	JCP2022_900005	37	ACADVL	ccsbBroad304_00007	trt
5	5	JCP2022_900006	41	ASIC1	BRDN0001485349	trt
6	6	JCP2022_900006	41	ASIC1	BRDN0001480309	trt
7	7	JCP2022_900006	41	ASIC1	ccsbBroad304_00008	trt
8	8	JCP2022_900007	47	ACLY	ccsbBroad304_00009	trt
9	9	JCP2022_900008	48	ACO1	ccsbBroad304_00010	trt
10	10	JCP2022_900009	52	ACP1	ccsbBroad304_00011	trt
11	11	JCP2022_900010	54	ACP5	ccsbBroad304_00012	trt
12	12	JCP2022_900011	56	ACRV1	ccsbBroad304_00013	trt
13	13	JCP2022_900012	59	ACTA2	ccsbBroad304_00014	trt
14	14	JCP2022_900013	70	ACTC1	ccsbBroad304_00015	trt
15	15	JCP2022_900014	81	ACTN4	ccsbBroad304_00016	trt
16	16	JCP2022_900015	88	ACTN2	ccsbBroad304_00017	trt

normal

To summarise the available data

Perturbation	Match perturbations	Distinctive features	Images
Overexpression	orf	orf_feature	orf_gallery
Knocked-Out	crispr	WIP	crispr_gallery
Compound	WIP	WIP	compound_gallery

For example:
- broad.io/orf
- broad.io/orf_feature
- broad.io/orf_gallery

Additionally, a table with all the available gene and their other ids:
- broad.io/babel

`broad_babel`: Obtain metadata

Translates identifiers and provides essential metadata.

- What is the NCBI id of this gene?
- Is X perturbation a treatment or a control?

It doubles as a single source of metadata ground truth.

`jump_portrait`: Spice-up your workflow with cell images

Fetch a subset of images associated to a perturbation.
Optionally, include their respective negative controls.

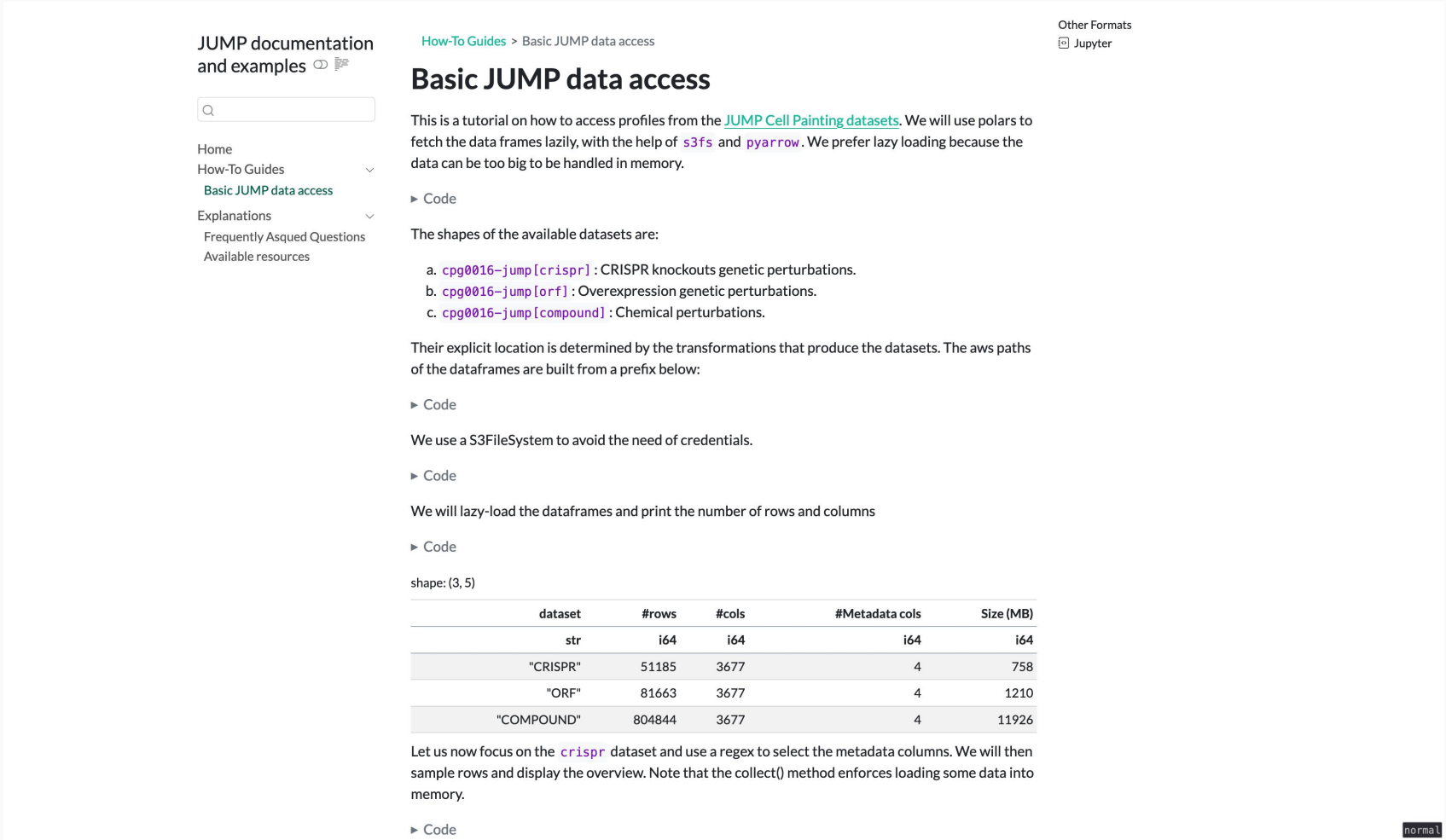
- Images at the site level
- Include plate-specific negative controls, to account for batch effects.
- Can be used to train ML/DL models on images by-request.

Other nice JUMP-adjacent tools

- jump-dti: Aggregate drug-target interaction databases
- cpg-data: Fetch images of Cell Painting Gallery

Final remarks

JUMP central concentrates knowledge



JUMP documentation and examples

Home
How-To Guides
Basic JUMP data access
Explanations
Frequently Asqued Questions
Available resources

How-To Guides > Basic JUMP data access

Other Formats
Jupyter

Basic JUMP data access

This is a tutorial on how to access profiles from the JUMP Cell Painting datasets. We will use polars to fetch the data frames lazily, with the help of `s3fs` and `pyarrow`. We prefer lazy loading because the data can be too big to be handled in memory.

▸ Code

The shapes of the available datasets are:

a. `cpg0016-jump[crispr]` : CRISPR knockouts genetic perturbations.
b. `cpg0016-jump[orf]` : Overexpression genetic perturbations.
c. `cpg0016-jump[compound]` : Chemical perturbations.

Their explicit location is determined by the transformations that produce the datasets. The aws paths of the dataframes are built from a prefix below:

▸ Code

We use a S3FileSystem to avoid the need of credentials.

▸ Code

We will lazy-load the dataframes and print the number of rows and columns

▸ Code

shape: (3, 5)

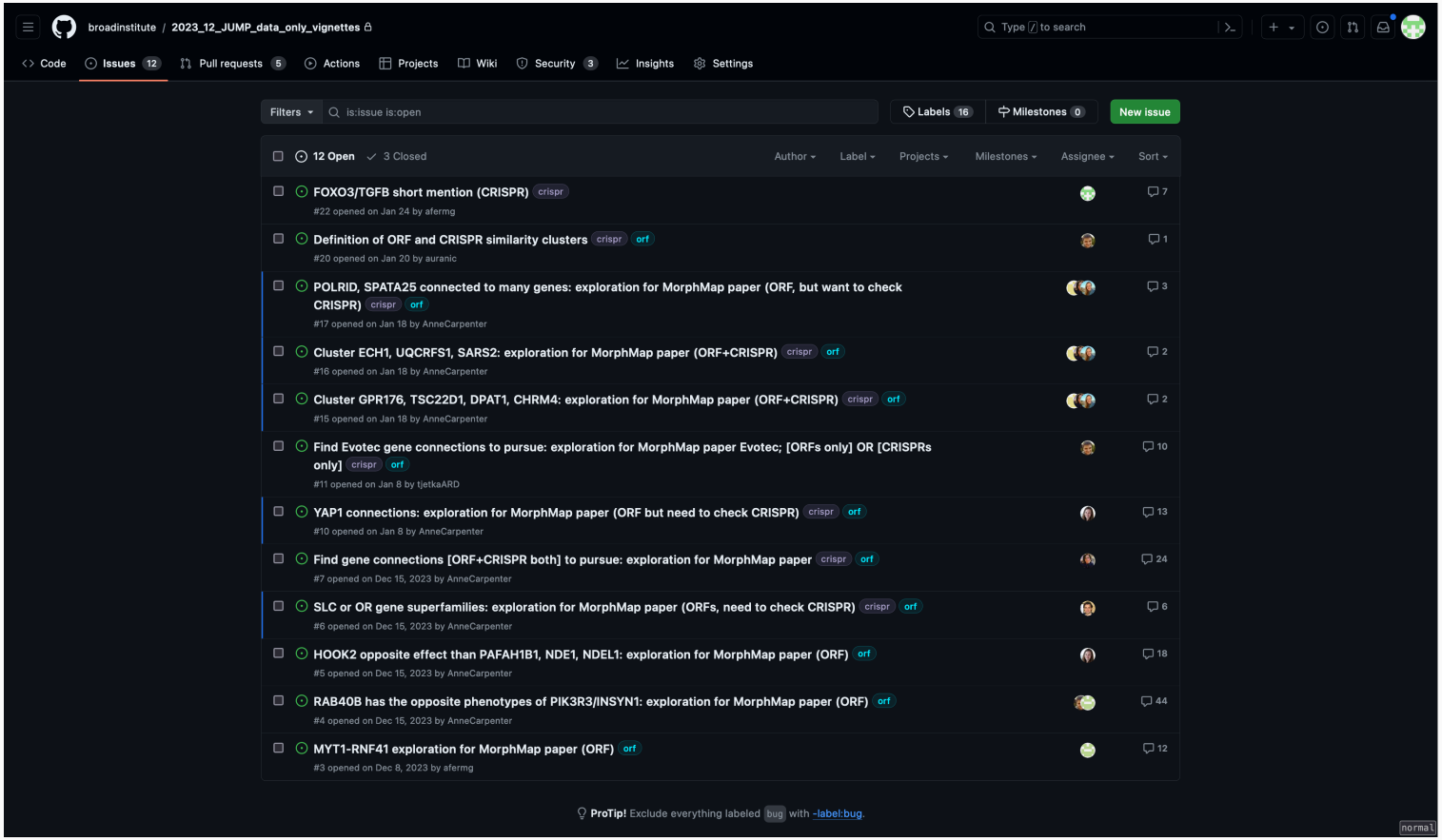
dataset	#rows	#cols	#Metadata cols	Size (MB)
str	i64	i64	i64	i64
"CRISPR"	51185	3677	4	758
"ORF"	81663	3677	4	1210
"COMPOUND"	804844	3677	4	11926

Let us now focus on the `crispr` dataset and use a regex to select the metadata columns. We will then sample rows and display the overview. Note that the collect() method enforces loading some data into memory.

▸ Code

normal

We are working on publishing biological vignettes to showcase the use of morphological data

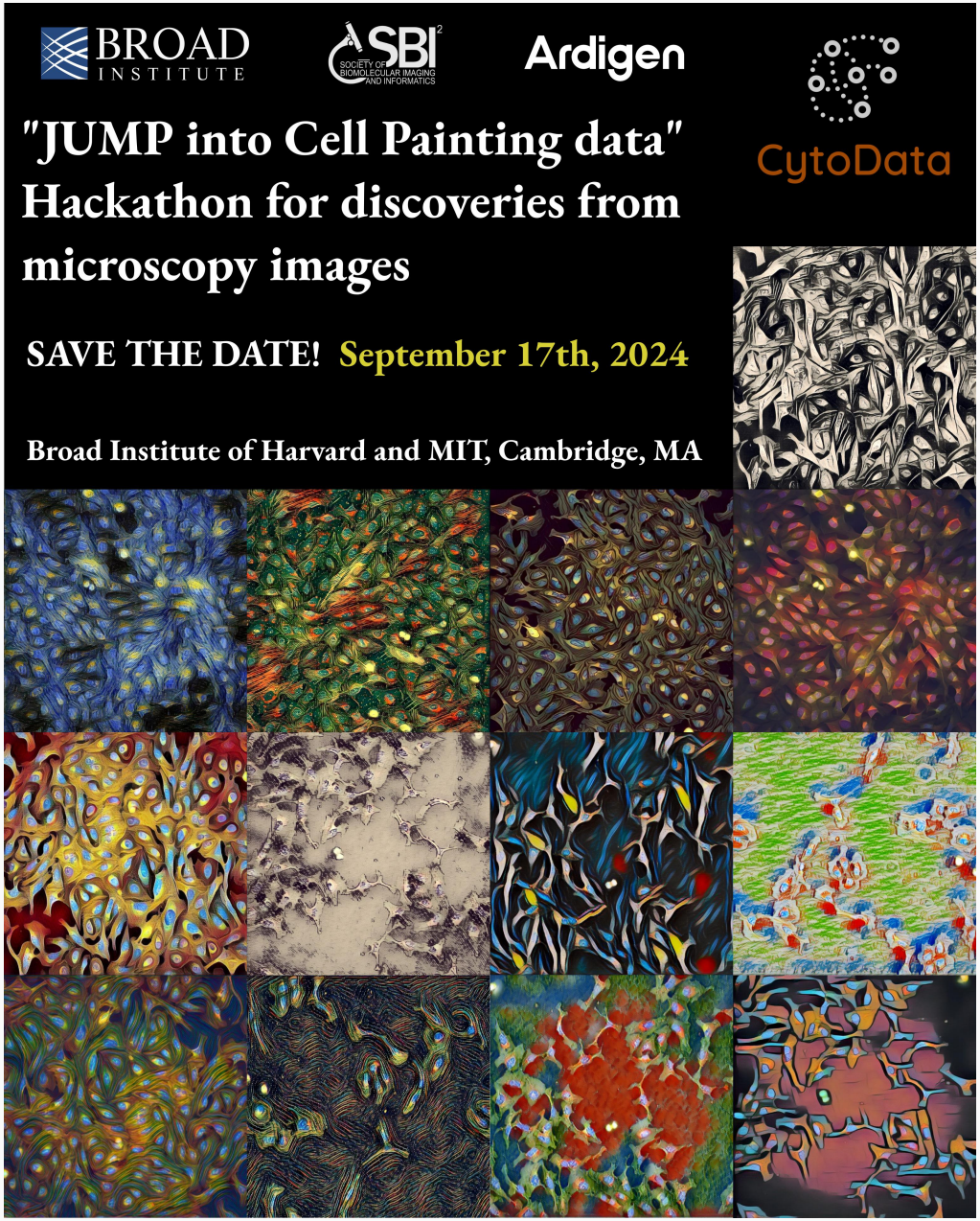


Conclusions

We built tools to solve our challenges and help scientists:

- Web tools for experimentalists to explore their perturbations of interest
- Python libraries for software inclined folks to integrate into their workflows
- A general framework to facilitate sharing results and learning material

Shameless plug: September JUMP Hackathon



Resources

- JUMP CP Consortium: jump-cellpainting.broadinstitute.org
- JUMP information central: broad.io/jump
- Imaging Platform's monorepo: broad.io/monorepo
- Slides: github.com/afermg/2024_05_JUMPTools_CellCircuits

Acknowledgements

Carpenter-Singh Lab

- Anne carpenter
- Shantanu Singh
- Niranj Chandrasekaran
- John Arevalo
- Sam Chen
- Ankur Kumar
- Ellen Su
- Alex Kalinin
- Adit Shah

BROAD INSTITUTE

Partners

Supporting partners

Collaborators

Scientific:
Horizon Discovery,
a PerkinElmer Company
Administrative:
Pistoia Alliance
Academic:
Umeå University
Stanford Machine Learning Group