https://github.com/fastplotlib/fastplotlib

fastplotlib

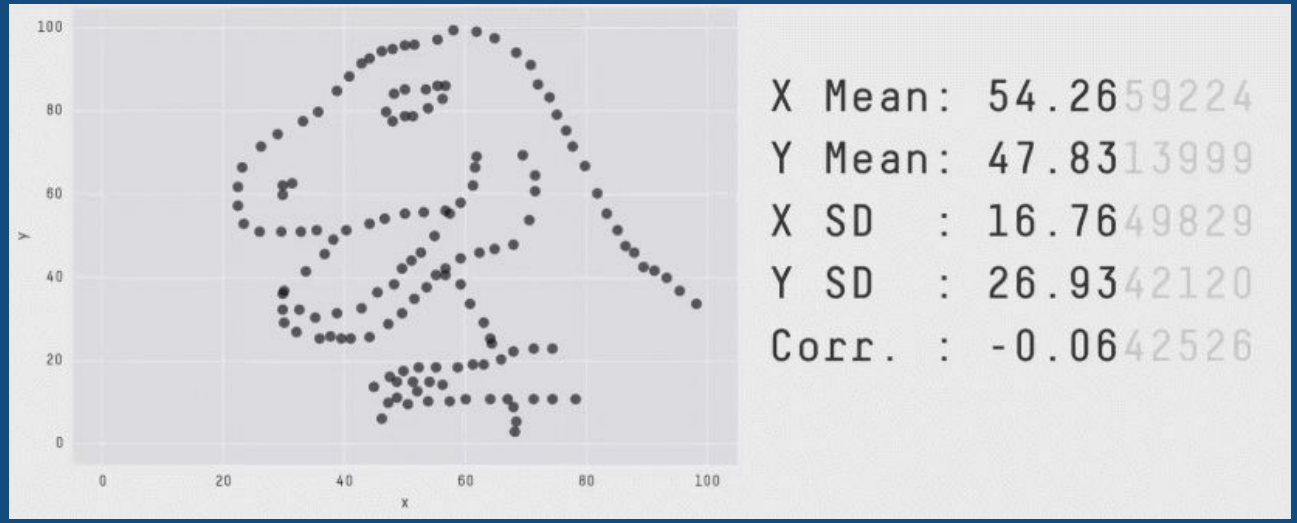
Ultrafast interactive visualizations

Caitlin Lewis **@caitlinllewis** **clewis7**

Kushal Kolar **@kushalkolar** **kushalkolar**

It is important to look at your data!

- Statistics are not sufficient
- “All models are wrong, some are useful”
- All algorithms are approximations

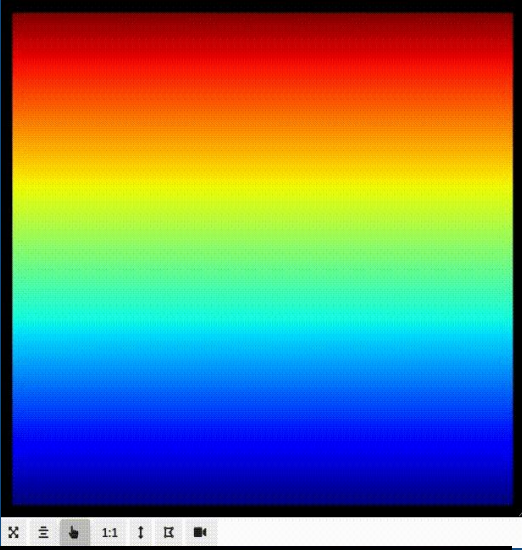


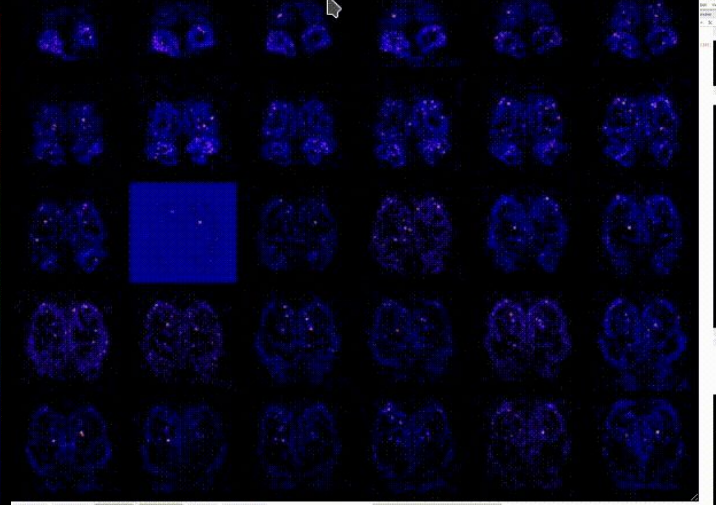
Matejka, Justin, and George Fitzmaurice. "Same stats, different graphs: generating datasets with varied appearance and identical statistics through simulated annealing." *Proceedings of the 2017 CHI conference on human factors in computing systems*. 2017.

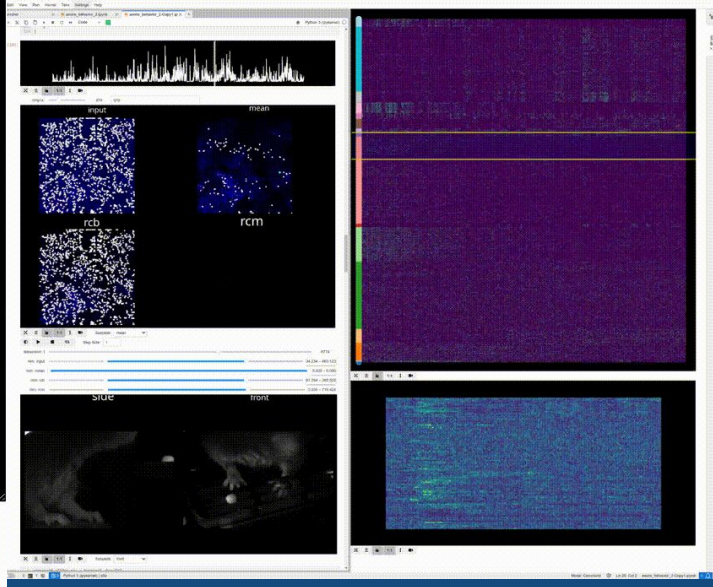
Why don’t scientific plots look as good as modern games?

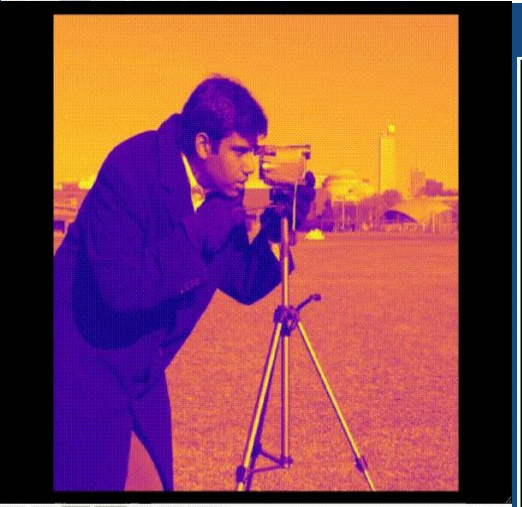
Graphics ~20 years ago

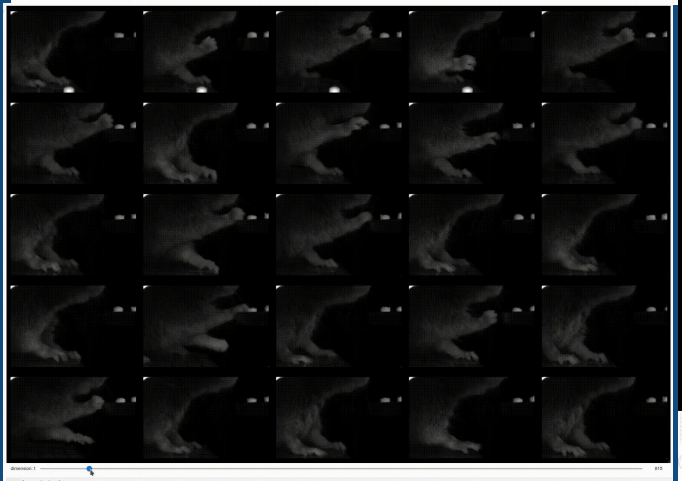
Current graphics











1:1
Subplots:
(0, 0)
Step Size:
1
Speed
100
dimension: t
0
1:1
1:1
0

fastplotlib-pygfx-wgpu Stack

Task

Image

fastplotlib

pygfx

wgpu

hardware

fastplotlib-pygfx-wgpu Stack

Image

```
fig = fpl.Figure() # create a figure

data = iio.imread("imageio:astronaut.png") # data

fig[0, 0].add_image(data=data) # plot an image

fig.show() # show the figure :D
```

Task

fastplotlib

pygfx

wgpu

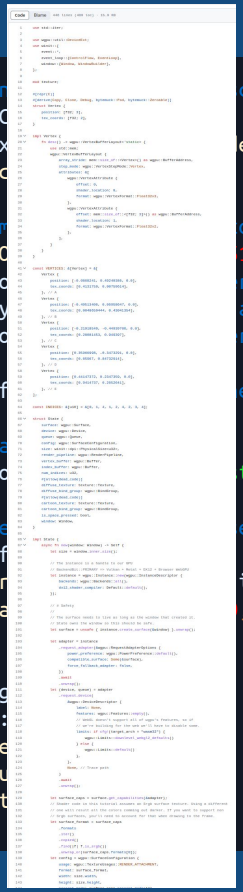
hardware

Core devs:

- Kushal Kolar
- Caitlin Lewis
- Almar Klein
- Amol Pasarkar

fastplotlib

fastplotlib-pygfx-wgpu Stack



```
# create a car                                    cene
canvas = WgpuC
renderer = gfx                                    erer(canvas)
scene = gfx.Sc

# create a car                                    o center of image
camera = gfx.O                                    12, 512)
camera.positio                                    r of the scene
camera.scale.y                                    axis
camera.positio                                    r of the scene

colormap1 = gf                                    e a colormap

# 512x512 arra
img_data = iio                                    tronaut.png").astype(np.float32) * 255

# define Geome                                    et as a Texture using the image data
image_obj = gf
    gfx.Geometr                                   img_data, dim=2)),
    gfx.ImageBa                                   , 255), map=colormap1),
)

scene.add(imag
def animate():
    renderer.re
    canvas.requ
canvas.request
canvas
```

Task

lotlib

gfx

gpu

ware

Core devs:

- Almar Klein
- Korijn van Golen

fastplotlib-pygfx-wgpu Stack

Task

Image

fastplotlib

pygfx

wgpu

hardware

- Vulkan
- Metal (Mac)
- DX12 (Windows)

New technologies: very fast, efficient, & leverage modern GPU hardware better than OpenGL

This is also what newer games use!

fastplotlib-pygfx-wgpu Stack

Task

Image

fastplotlib ~4 lines

pygfx ~15 lines - rendering engine

wgpu ~400 lines

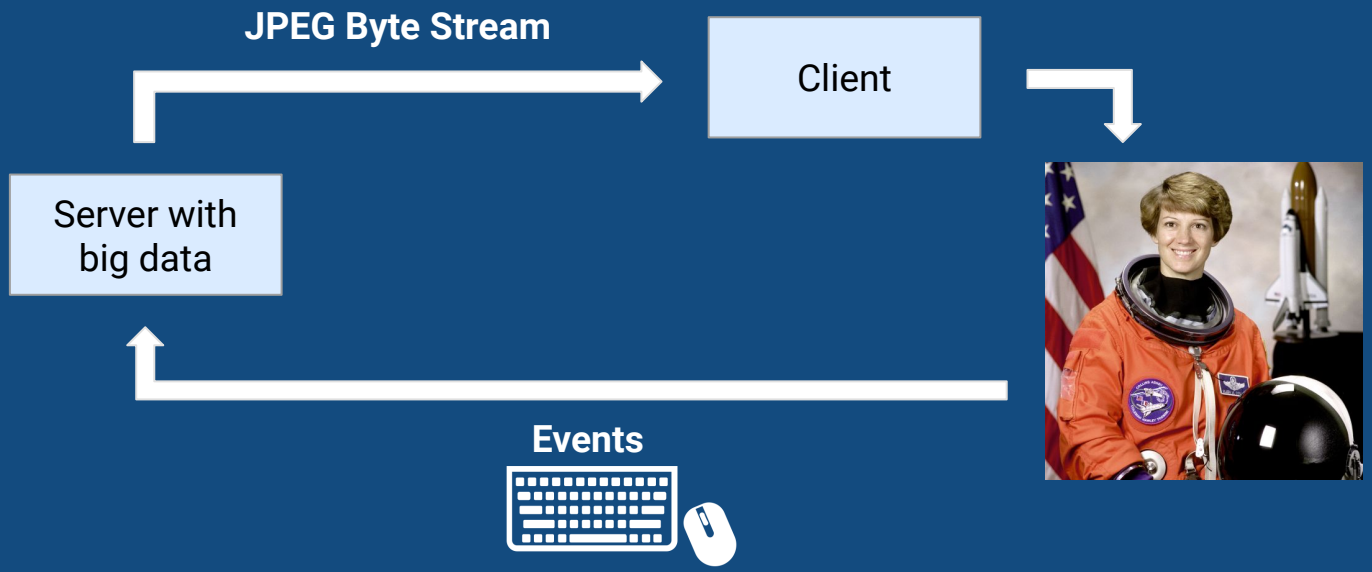
hardware ~700 lines

fastplotlib

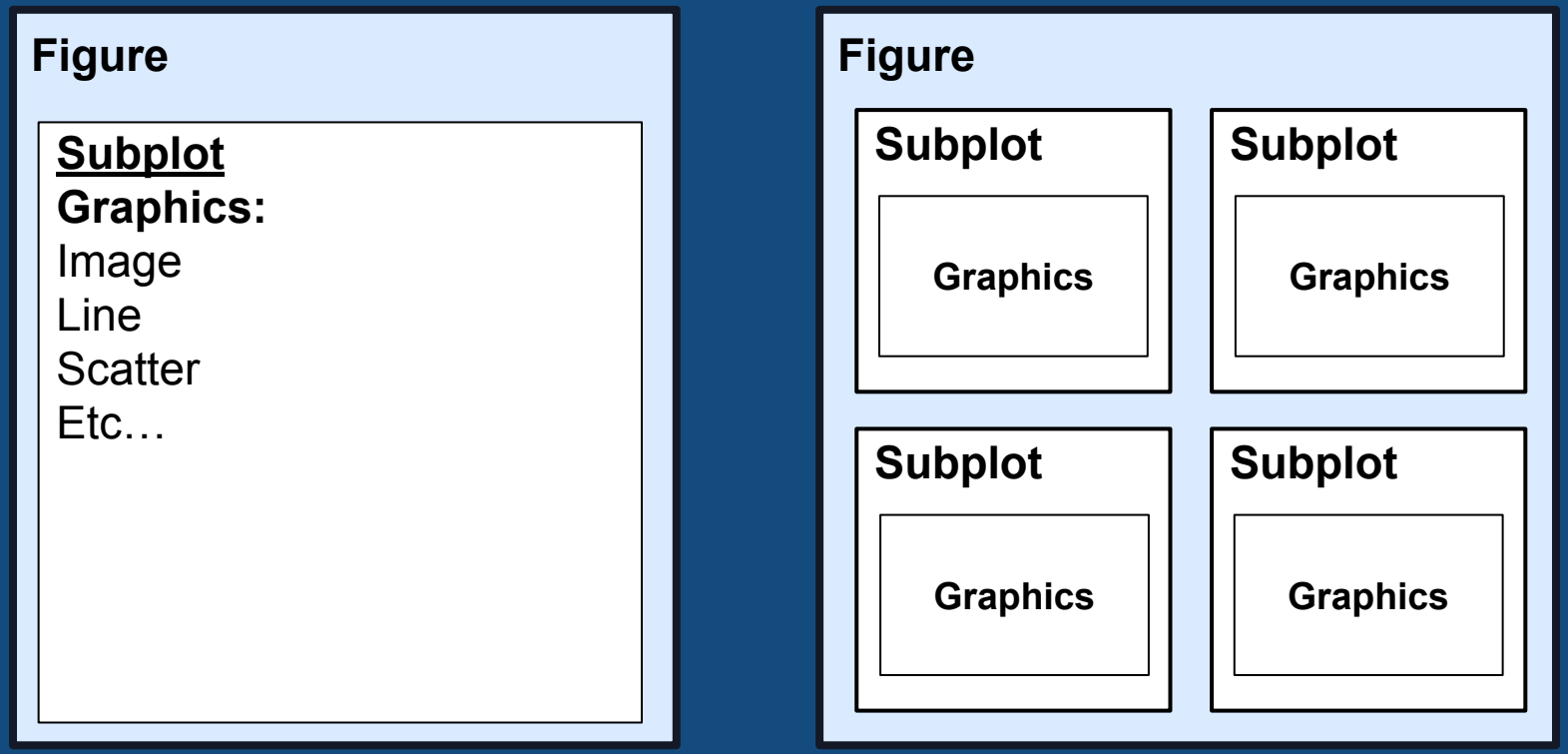
- High-level API for scientific plotting - inspiration from *pyqtgraph* and other libs
- Uses the *pygfx* rendering engine
- Very new - April 2022
- **Interactive in jupyter notebooks - cloud computing, remote infrastructure**
- Goals: fast visualization, **expressive & elegant API -** we'll tell you what this means!
- Core developers:
  - **Kushal Kolar - Flatiron/NYU**
  - **Caitlin Lewis - Duke University**
  - **Almar Klein - Independent/Flatiron**
  - **Amol Pasarkar - Columbia University**
- Possible *fastplotlib* backend for napari in the future :D

Fastplotlib via remote frame buffer

- **jupyter-rfb**
  - Server-side rendering, client only receives a jpeg byte stream
- Faster than client libs - bokeh, dash, plotly, etc.
  - Render big data on server/cloud, client only gets small jpeg stream!

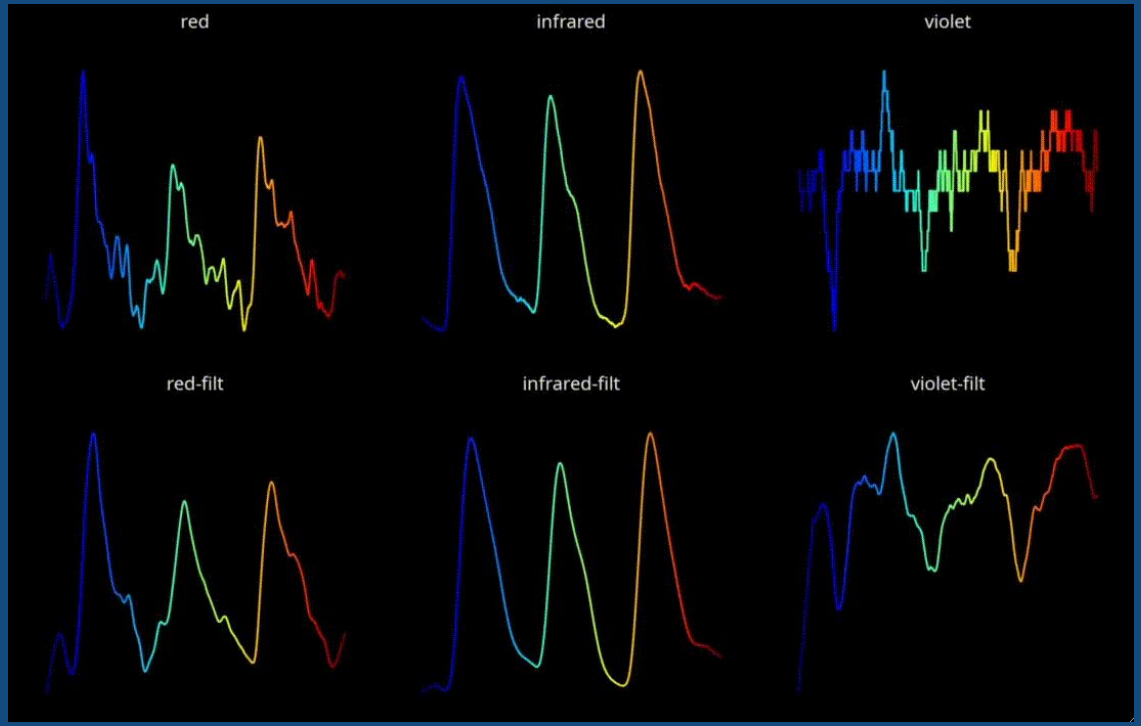


fastplotlib API

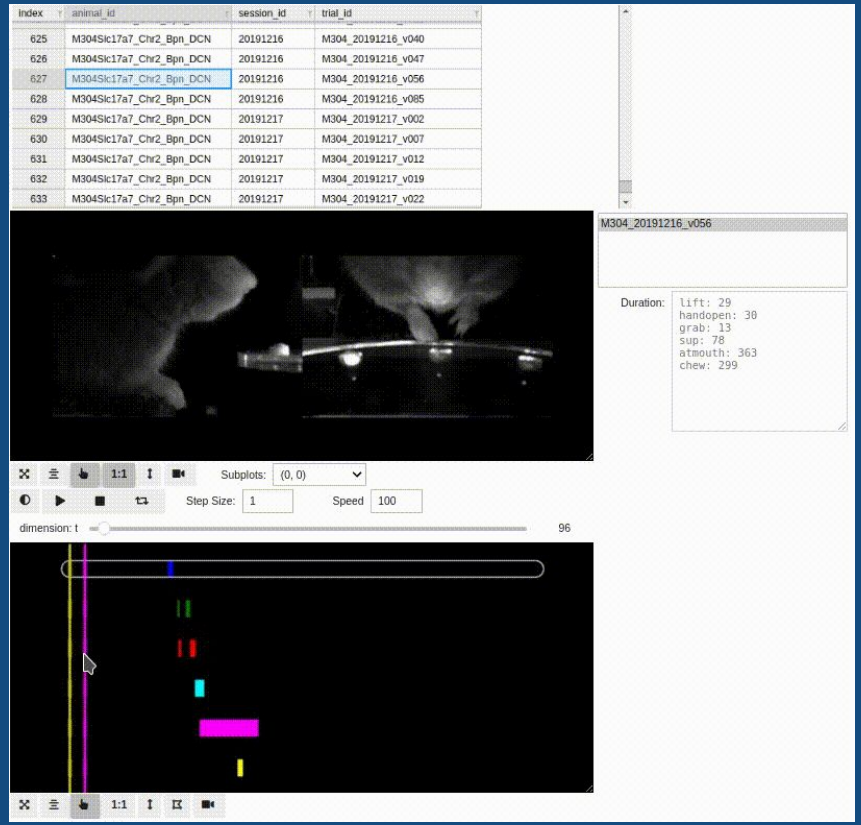


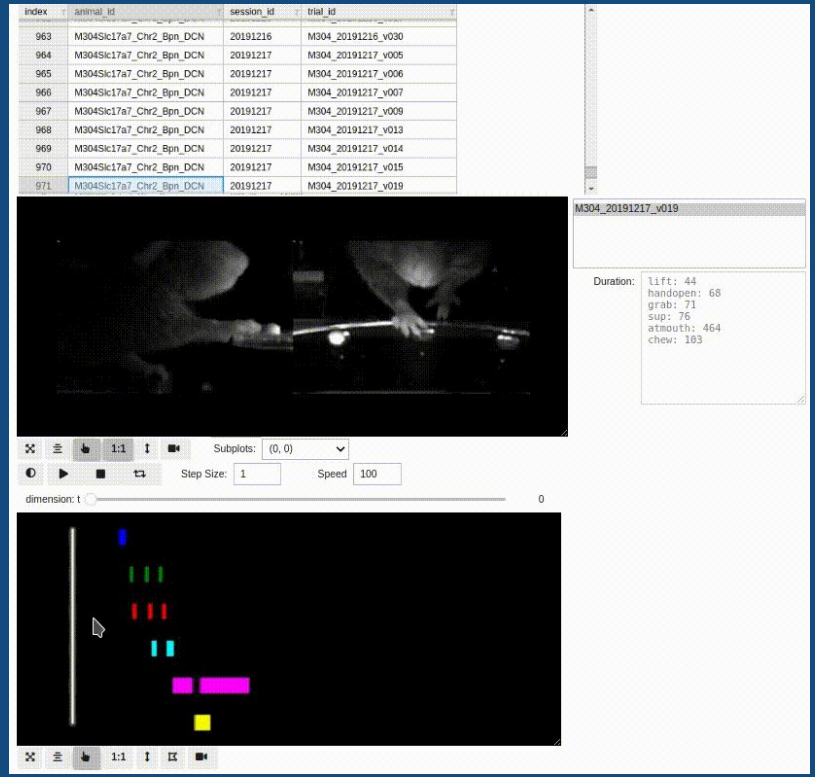
Example Applications

Real time sensor data - pulse ox development (Arjun)

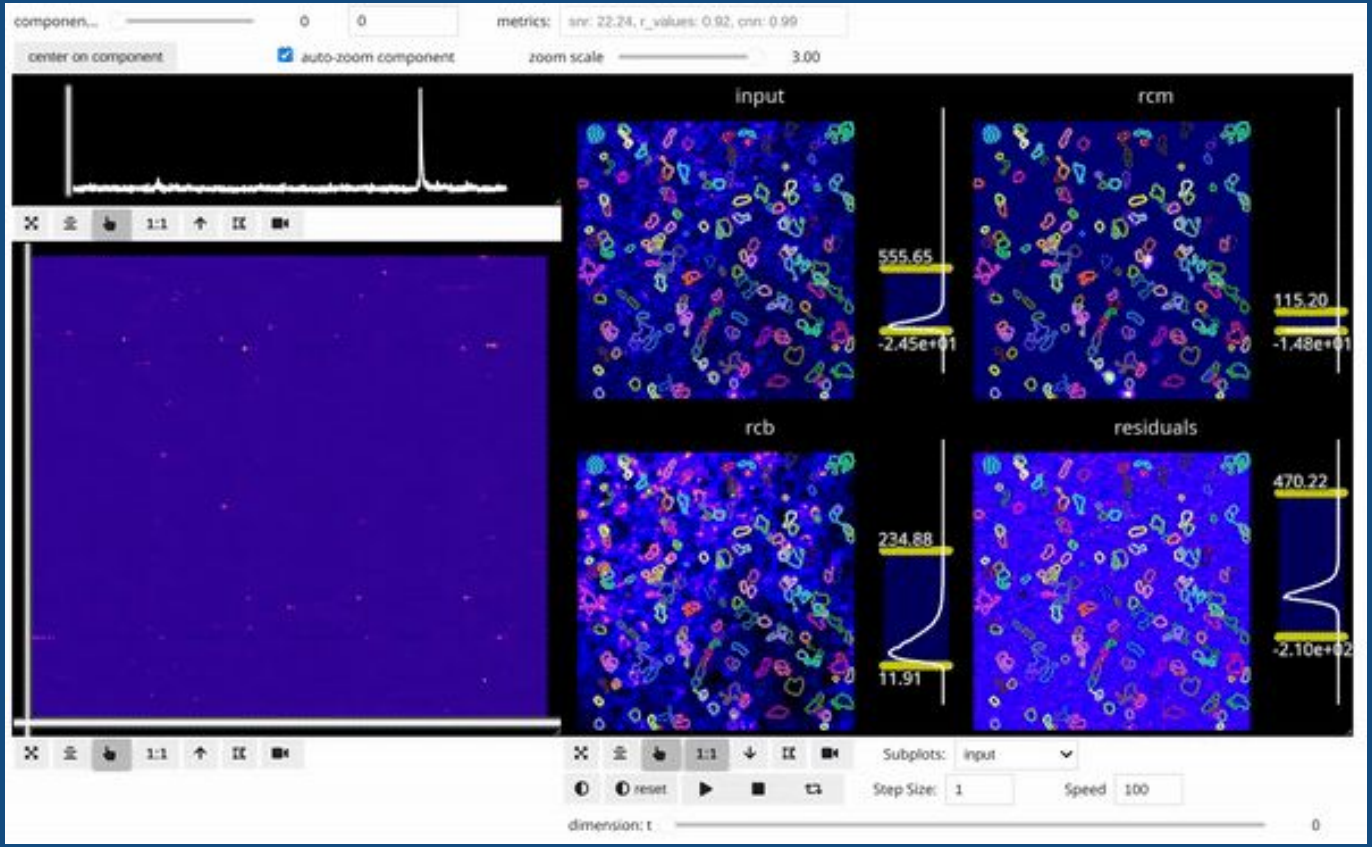


animal-soup


index
animal_id
session_id
trial_id
625 M304Slc17a7_Chr2_Bpn_DCN 20191216 M304_20191216_v040
626 M304Slc17a7_Chr2_Bpn_DCN 20191216 M304_20191216_v047
627 M304Slc17a7_Chr2_Bpn_DCN 20191216 M304_20191216_v056
628 M304Slc17a7_Chr2_Bpn_DCN 20191216 M304_20191216_v085
629 M304Slc17a7_Chr2_Bpn_DCN 20191217 M304_20191217_v002
630 M304Slc17a7_Chr2_Bpn_DCN 20191217 M304_20191217_v007
631 M304Slc17a7_Chr2_Bpn_DCN 20191217 M304_20191217_v012
632 M304Slc17a7_Chr2_Bpn_DCN 20191217 M304_20191217_v019
633 M304Slc17a7_Chr2_Bpn_DCN 20191217 M304_20191217_v022
M304_20191216_v056
Duration:
lift: 29
handopen: 30
grab: 13
sup: 78
atmouth: 363
chew: 299
1:1
Subplots: (0, 0)
Step Size: 1
Speed 100
dimension: t
96
1:1


index
animal_id
session_id
trial_id
963 M304Slc17a7_Chr2_Bpn_DCN 20191216 M304_20191216_v030
964 M304Slc17a7_Chr2_Bpn_DCN 20191217 M304_20191217_v005
965 M304Slc17a7_Chr2_Bpn_DCN 20191217 M304_20191217_v006
966 M304Slc17a7_Chr2_Bpn_DCN 20191217 M304_20191217_v007
967 M304Slc17a7_Chr2_Bpn_DCN 20191217 M304_20191217_v009
968 M304Slc17a7_Chr2_Bpn_DCN 20191217 M304_20191217_v013
969 M304Slc17a7_Chr2_Bpn_DCN 20191217 M304_20191217_v014
970 M304Slc17a7_Chr2_Bpn_DCN 20191217 M304_20191217_v015
971 M304Slc17a7_Chr2_Bpn_DCN 20191217 M304_20191217_v019
M304_20191217_v019
Duration:
lift: 44
handopen: 68
grab: 71
sup: 76
atmouth: 464
chew: 103
1:1
Subplots: (0, 0)
Step Size: 1
Speed 100
dimension: t
0
1:1

mesmerize-viz



Current state of fastplotlib

- Late Alpha
  - Some things are still evolving - we are constantly improving things!
  - Don’t hesitate to post an issue or discussion forum post!
- Moderate test coverage
  - ~90%: Graphics, graphic features
  - ~70% layouts
  - ~20%: selector tools
- Some basic components are not ready yet
  - Axes are coming very soon!

Roadmap for 2025

Contributions and ideas are welcome from people with all levels of experience! :D

There are several items highlighted with 🟢 that are perfect for newcomers!

https://github.com/fastplotlib/fastplotlib/issues/55

Thanks :)