

Implementing Algorithms for Expanded Functionality of Spatial Single Cell Analysis FNL/Datamine

Team Members: Aileen Chow, Ahmad Abdallah, Arshnoor Randhawa, Ella Delvecchio, Jiazhen Li, Liza Shchelik, Megan Lawson, Mustapha Braimoh, Ruhi Sharmin, Sam Ying, Shamita Yedlapalli
TAs: Alex Liu, Omar Eldaghar, Thomas Sheeley



What is SPAC?

SPAC is a python library used to analyze spatial single-cell datasets derived from cancerous tumors which incorporates existing data analysis tools into a single package for ease of use. SPAC utilizes the Shiny platform to generate a front-end UI that allows researchers to visualize these datasets.

SPAC RealTime Visualization Layer

SPAC Interactive Analysis Layer

SPAC Python Package

Dependencies

- Scalable **RealTime** visualizations/annotations (WIP)
- Vitesse + Voila

- NIDAP CodeWorkbook templates and Pipelines
- Scalable on NIDAP + HPC **GPUs** (Biowulf/FRCE)
- Technical free reports to share with collaborators

- Scientific computation, analysis methods, R&D tools
- Unit tests, meaningful error messages
- Standard python documentation

- AnnData, scanpy
- squidpy, scimap
- plotly, matplotlib, Vitesse

Project Goals

- Expand functionality of the front-end interface
 - Customization of parameter options within each visualization
 - UI elements for adjusting styling elements of generated plots
 - Novel features which allow flexible subsetting and data wrangling
- Optimization of SPAC visualizations
 - Characterizing the SPAC visualization run times
 - Refactoring existing functions to utilize optimal computational methods
- Implementation of clustering algorithms
 - K-Nearest Neighbors
 - K-Means
 - Phenograph GPU acceleration

User Experience

SPAC Interactive Dashboard

Choose a file to upload:

Browse...

dev_example.pickle

Upload complete

of Rows: 4825

of Columns: 5

Obs: broad_cell_type, phenograph_0_35, renamed_phenotypes, phenograph_0_6

Obsm: spatial, X_umap

Layers: arcsinh, arcsinh_z_scores

Uns: phenograph_features

Data InputAnnotationsFeaturesBoxplotFeat. Vs AnnoAnno. Vs AnnoSpatialUMAPScatterplot

SPAC Interactive Dashboard

Choose a file to upload:

Browse...

dev_example.pickle

Upload complete

Number of Cells: 4825

Associated Tables: spatial, X_umap

Number of Features: 5

Tables: arcsinh, arcsinh_z_scores

Annotations: broad_cell_type, phenograph_0_35, renamed_phenotypes, phenograph_0_6

Unstructured Data: phenograph_features

☐ Subset Annotation

Subset Data

Restore Original Data

No subsets have been made yet.

SPAC Terminology

SPAC uses general terminology to simplify technical terms from the AnnData object for less technical users. Here is a quick guide:

- Cells:** Rows in the X matrix of AnnData.
- Features:** Columns in the X matrix of AnnData, representing gene expression or antibody intensity.
- Tables:** Originally called layers in AnnData, these represent transformed features.
- Associated Tables:** Corresponds to .obsm in AnnData and can store spatial coordinates, UMAP embeddings, etc.
- Annotation:** Corresponds to .obs in AnnData and can store cell phenotypes, experiment names, slide IDs, etc.

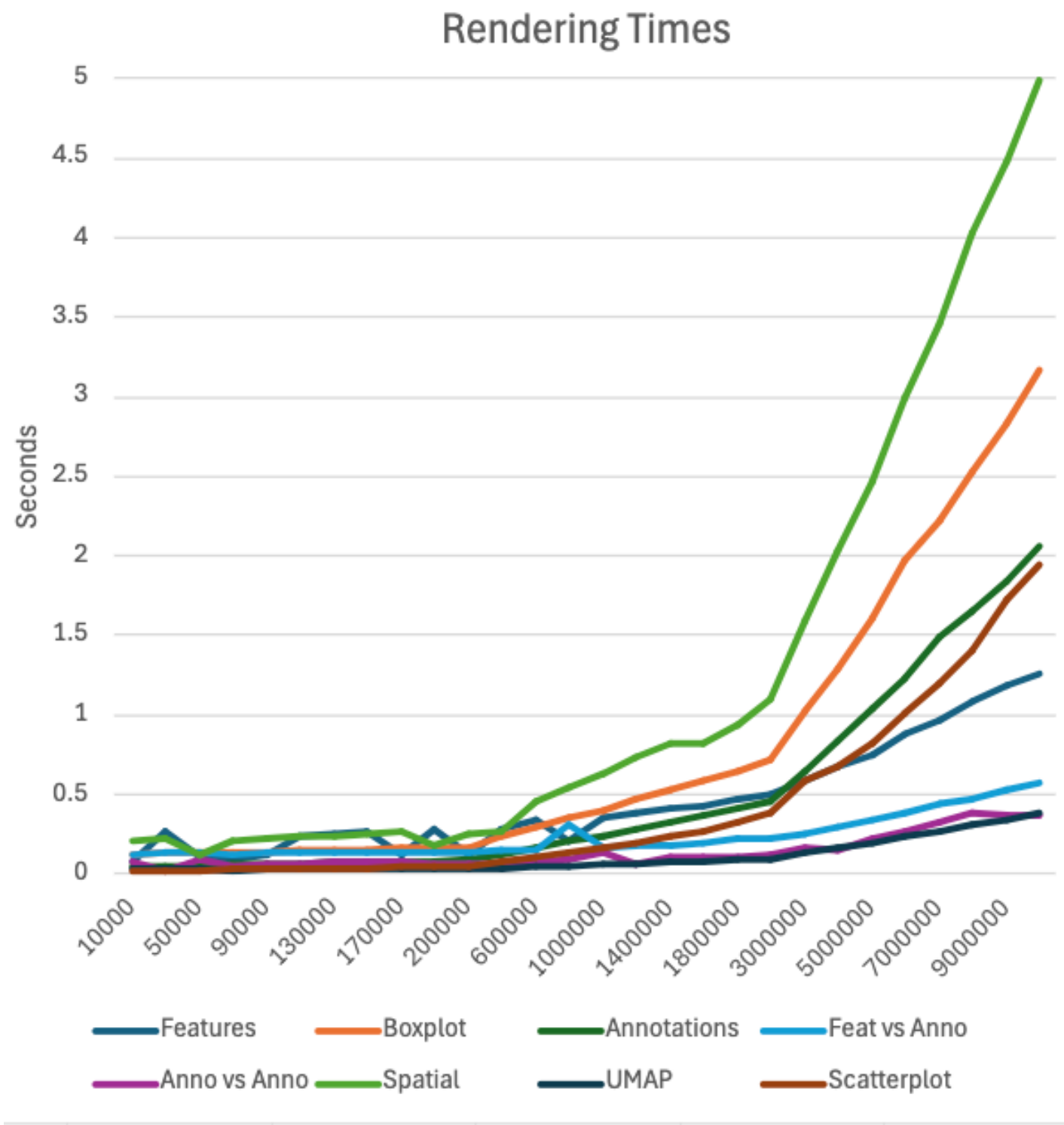
For more in-depth explanations, visit our [GitHub page](#).

Original SPAC Interactive Dashboard

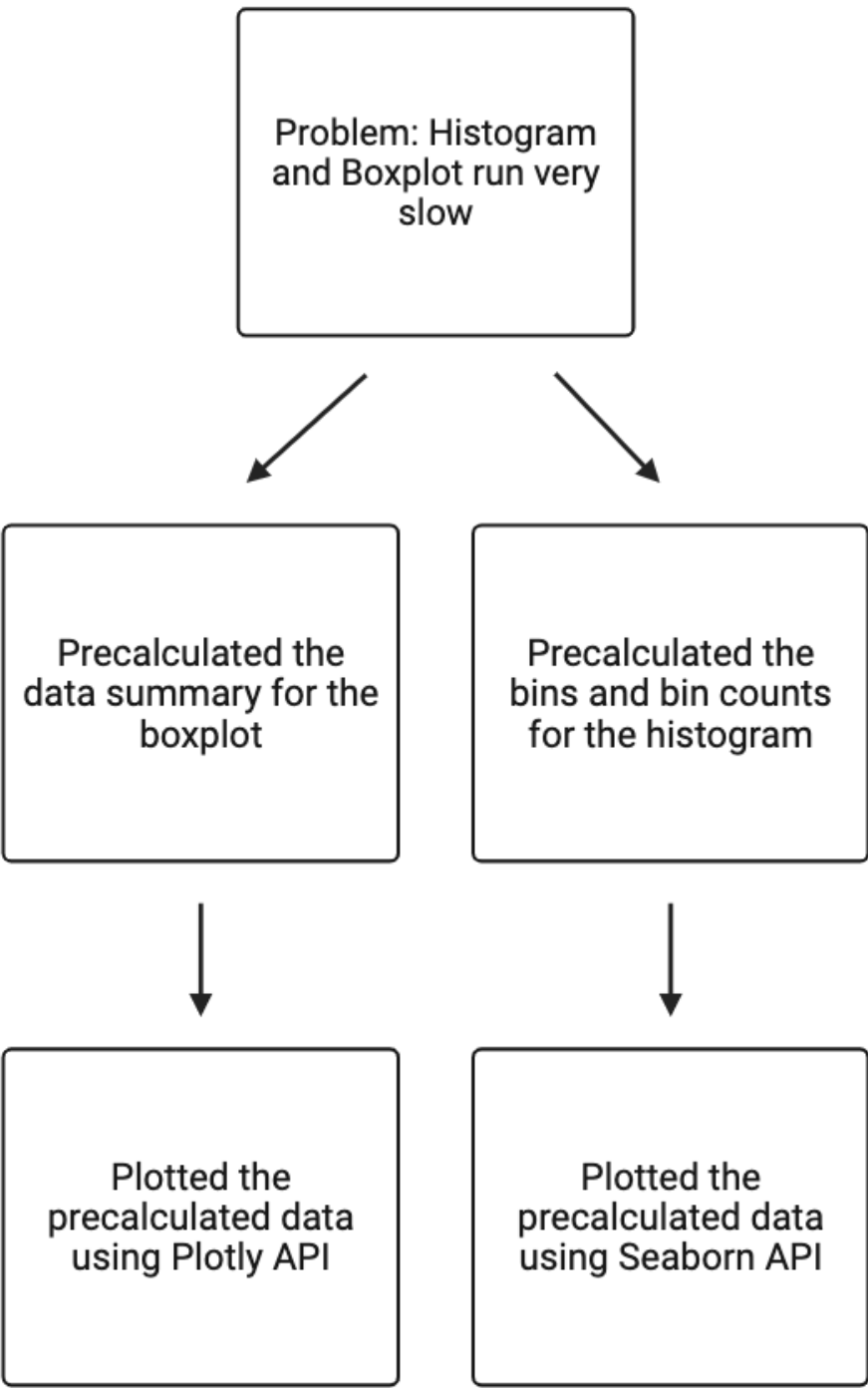
Improved Interactive SPAC Dashboard

Realtime Layer Optimization

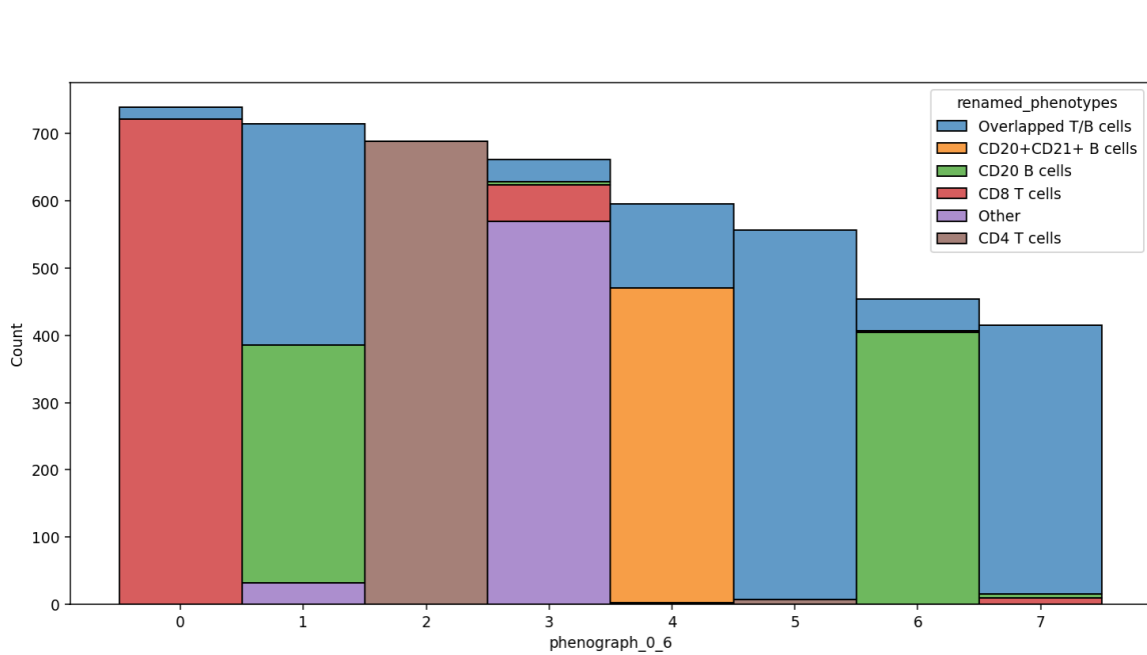
Visualization Runtimes



Implementing Optimizations



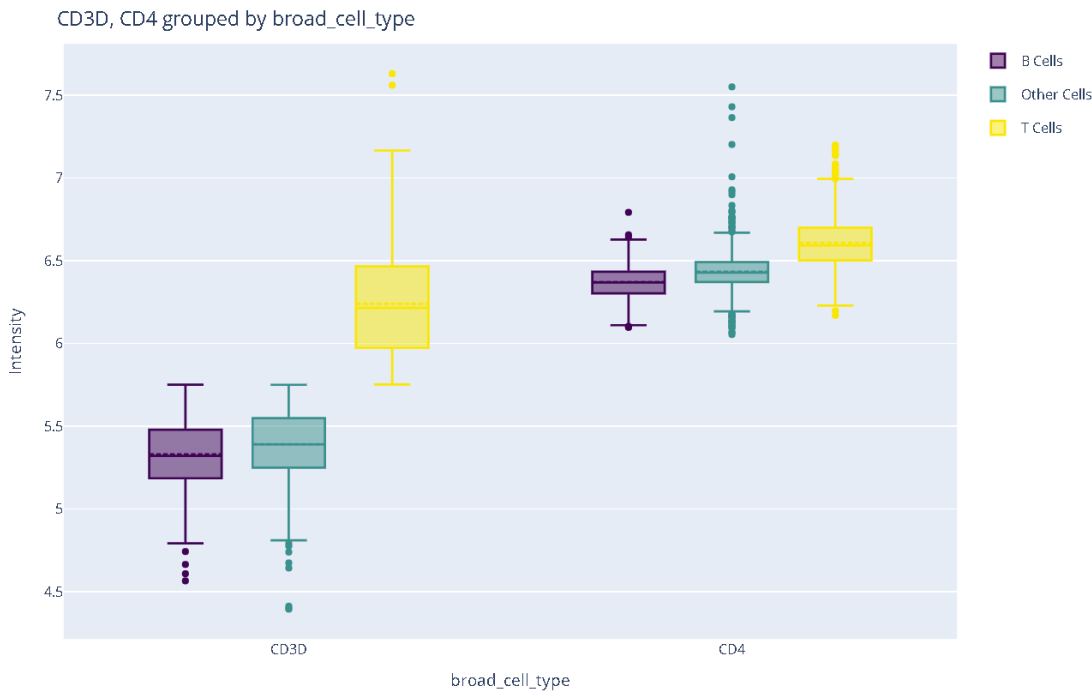
Histogram Results



Histogram (10M cells with annotations):

- Original Implementation: 35.24 s
- New Implementation: 2.56 s
- 1376%** increase in efficiency

Boxplot Results



Boxplot (10M cells with annotations):

- Original Implementation: 20.91s
- New Implementation: 2.28s
- 917%** increase in efficiency

Conclusions

- Expanded the functionality of the front end interface by adding nearly 50 UI features.
- Enabled more dynamic data subsetting and display in front end UI.
- Implemented two clustering algorithms to the SPAC python package.
- Optimized existing boxplot and histogram visualization functions by nearly 10x.
- Laid ground work for Phenograph GPU acceleration

Acknowledgements

We'd like to thank our TAs: Thomas Sheeley, Alex Liu, Omar Eldaghar, our FNL mentor George Zaki, Fang Liu, Rui He, and the Data Mine staff.