

1 Results

All of our works are fully open source and available in this GitHub Repo with detailed Readme.md instructions.

1.1 System Implementation

In this section, we will introduce in detail the implementation and test results of several key functions and project objectives in the Cellxgene VIP extension, including Dataset Creation Pipeline, Muon Data Adaptor, Dual Mappings View, and ATAC Tracks View.

1.1.1 Data Creation Pipeline

The Dataset Creation Pipeline is designed to help users generate single-cell multiome data. Its main function is to integrate different types of single-cell data into a unified format Muon (.h5mu), so that our Cellxgene variants can accept the data for subsequent analysis and visualization. In this process, we implemented steps such as data filtering, standardization, and feature selection, and ensured that these steps can handle different types of data (such as RNA-seq and ATAC-seq). In addition, we also implemented data format conversion and finally output the processed data into Muon format. This pipeline is largely follow's Muon tutorial's PBMC 3k Cell process notebook[1] but with different mappings choices. Code samples are shown in Fig 1.

```
PCA / TSNE Mapping

PCA

In [ ]: sc.tl.pca(rna, svd_solver='arpack')

In [ ]: sc.pp.neighbors(rna, n_neighbors=10, n_pcs=20)

In [ ]: sc.tl.leiden(rna, resolution=.5)

TSNE

In [ ]: sc.tl.tsne(rna)

Save Data

In [ ]: mdata.write("muon_data/pbmc10k.h5mu")
```

Fig. 1. Code of Generating Mappings for RNA-seq Data

The pipeline also has the function of converting the generated Muon data into ATAC Tracks data, which can help users easily generate the ATAC Tracks data they need and upload it to their own server for use in the ATAC Tracks View interface. This part of the pipeline requires an additional

Chrom Size file, which the user needs to select and download. Code samples are shown in Fig 2.

```
In [ ]: import numpy as np
import muon as mu
from scipy.ndimage import gaussian_filter1d
import pyBigWig
import pandas as pd
import scipy.ndimage

In [ ]: # read your muon data here
mdata_processed = mu.read("muon_data/pbmc10k_processed.h5mu")
mdata_processed

atac = mdata_processed.mod['atac']
atac

counts = atac.layers['counts'].toarray()
intervals = atac.var['interval']

In [ ]: smooth_sigma = 2
smoothed_counts = scipy.ndimage.gaussian_filter1d(counts, sigma=smooth_sigma, axis=1)

In [ ]: def smooth_counts(counts, sigma=3):
    return gaussian_filter1d(counts, sigma=sigma, axis=0)

smoothed_counts_t = smooth_counts(counts)

In [ ]: smoothed_counts.shape
```

Fig. 2. Code of Smooth Functions of ATAC Tracks Data

All Dataset Creation Pipeline codes are open source and published in the project's GitHub Repo, we have two folders separately named *muon_tutorial* and *atac_pipeline*, as shown in Fig 3. Generating Muon Data covers three Jupyter Notebook files, which are processing RNA-seq data, processing ATAC-seq data, and dual data integration. Users will execute these three files in a fixed order to obtain the final Muon data file. Generating ATAC Track Data only requires running a Jupyter Notebook file, which will generate the corresponding ATAC Track Data from the Muon data and Chrom Size files.












 YourHarbour	Update install_VIPlight.sh	fa88987 · 3 days ago	 714 Commits	▼
 DataTables	v1.0.2, please execute update.v1.0.2.sh to update scanpy	4 years ago		
 ace	CLI draft	4 years ago		
 atac_pipeline	Add atac_track_process notebook	last month		
 bin	add slim the h5ad for plotting functions	2 months ago		
 d3plot	Keep font size of gene names constant in iVolcano plot	4 years ago		
 env.yml	Update VIPlight.yml	last month		
 gsea	add GSEA gmt files	3 years ago		
 jspanel	v1.0.2, please execute update.v1.0.2.sh to update scanpy	4 years ago		
 muon_tutorial	Add muon data pre-process pipeline	last month		

Fig. 3. Files Structure on GitHub

1.1.2 Muon Data Adaptor

Muon Data Adaptor is an important part of Cellxgene development, which is responsible for converting input Muon data into a unified format that can be used by Cellxgene.

During the implementation process, we focused on ensuring seamless conversion and integration of different data types to facilitate subsequent multi-dimensional data visualization. The adapter first loads data from the Muon file and performs necessary validation to ensure that the data format is correct and can be processed without errors. Then merge the RNA-seq and ATAC-seq data in the Muon dataset. The adapter integrates the two data by adding suffixes to the column names to distinguish RNA and ATAC data and integrating related metadata and mappings embedding. This ensures that the integrity of the two data types is maintained in the generated Cellxgene data.

In order to be compatible with the original Anndata Adaptor, we inherited most of the functions in the Anndata Adaptor Class, such as returning data structure, cell information, etc., and added RNA-seq and ATAC-seq data merging function, data validation and other functions on this basis to implement Muon Data Adaptor. Part of the code is shown in the Fig 4.

```
142 def _merge_muon_data(self, atac, gex):
143     gex.obs = gex.obs.add_suffix('_gex')
144     atac.obs = atac.obs.add_suffix('_atac')
145
146     gex.obsm = {f"{key}_gex": value for key, value in gex.obsm.items()}
147     atac.obsm = {f"{key}_atac": value for key, value in atac.obsm.items()}
148
149     gex.uns = {f"{key}_gex": value for key, value in gex.uns.items()}
150     atac.uns = {f"{key}_atac": value for key, value in atac.uns.items()}
151
152     assert np.array_equal(gex.obs_names, atac.obs_names)
153
154     merged = gex.copy()
155     merged.obsm.update(atac.obsm)
156
157     for col in atac.obs.columns:
158         merged.obs[col] = atac.obs[col]
159
160     self.data = merged
```

Fig. 4. Code of Merge Functions in Muon Data Adaptor

In the functional test, Muon Data Adaptor performed well, successfully processing Muon data that meets the specifications, and stopped and returned error messages in early stage when encountering wrong input file format or do not meet the specifications. The integrated data after parsing the Muon data file also performed stably in the subsequent Cellxgene visualization process.

1.1.3 Dual Mappings View

The goal of the dual mapping view is to allow multiple data mappings, such as RNA-seq and ATAC-seq t-SNE plots, to be displayed simultaneously in the same interface. This feature enables researchers to visually compare these datasets side by side, making it easier to identify correlations or differences between gene expression and chromatin accessibility. After modifying and adding the Muon Data Adaptor, we make some modifications to develop the Cellxgene front end for this feature. Through the development strategy mentioned in the method section, we successfully implemented the Dual Mappings View feature.

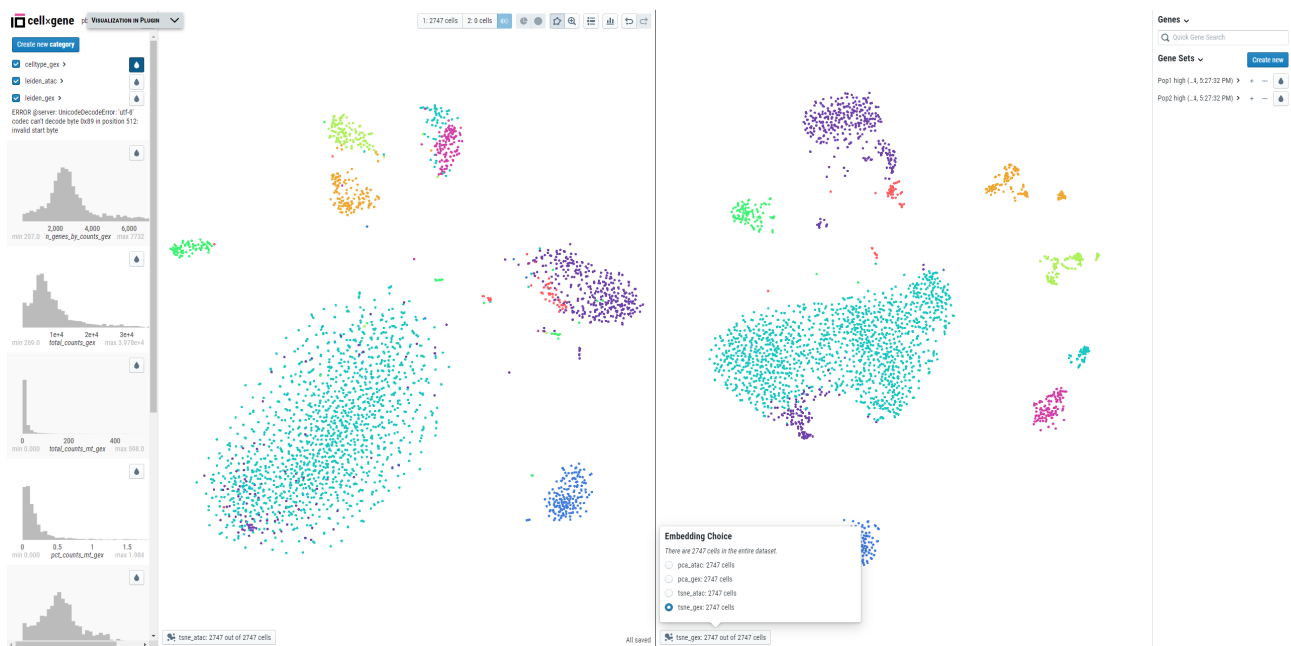


Fig. 5. Sample of Dual Mapping View of PBMC3k Dataset (Shows GEX t-SNE and ATAC t-SNE Mappings of Cells)

As shown in the Fig 5, users can load t-SNE data mappings of RNA-seq and ATAC-seq at the same time, which are displayed in separate but adjacent panels for direct visual comparison. The left panel still contains all the original functions, such as lasso tools, etc. The right panel is our newly added mappings display panel. In order to maximize compatibility and loading speed, we have cancelled most of the original functions and only retained the function of switching mappings. But our new panel will still synchronize some operations of the corresponding left panel, such as the synchronous highlighting function of clusters, which is very important to us.

When the user selects a cluster in one dataset, the corresponding cluster in the other dataset will be automatically highlighted. This synchronization is critical for understanding the behavior of certain cell populations in different omics layers. Fig 6 shows this feature in detail with an example, we highlight the same cluster of oligodendrocyte. By clicking the category button in the menubar above, the dual mappings automatically display different clusters. When the mouse moves and hover to a cluster in the RNA-seq t-SNE graph on the left, the system automatically highlights the corresponding cluster in the ATAC-seq graph on the right, thereby revealing how chromatin accessibility is associated with gene expression in a specific cell population.

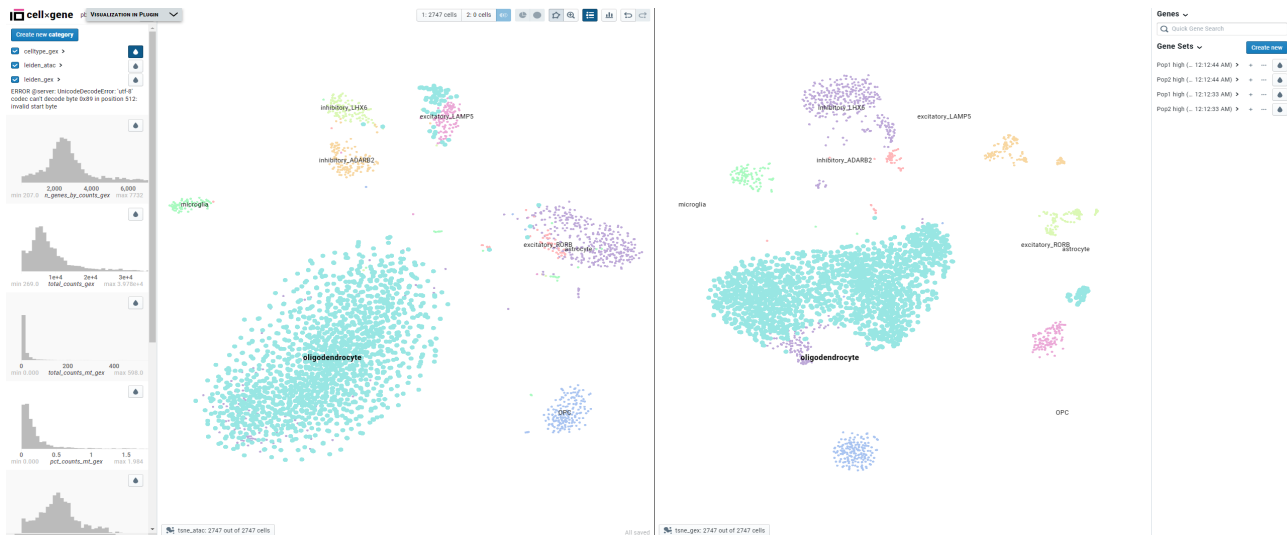


Fig. 6. Sample of Dual Mapping View Sync Highlighting of Clusters

1.1.4 ATAC Tracks View

We have successfully implemented the ATAC Tracks View feature in the Cellxgene VIP extension. This extension allows users to enter ATAC Tracks URLs through a graphical interface and dynamically load and display these tracks in the browser. The implementation process involves initializing the interface, handling user input, configuring the IGV.js browser, and managing the display of multiple ATAC tracks simultaneously.

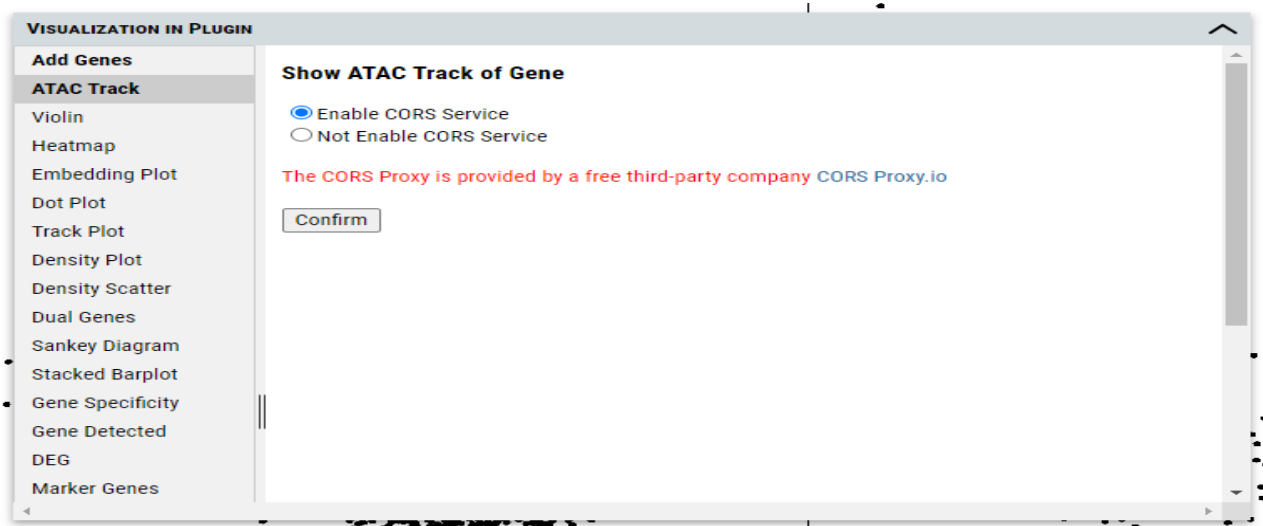


Fig. 7. CORS Choose Screen of ATAC Tracks View

The key goal was to ensure that ATAC Tracks View can correctly load multiple data sources, including those that require Cross-Origin Resource Sharing (CORS) support. This feature includes a screen to enable or disable CORS, depending on the user's requirements for the data sources, the flexibility enables researchers to access data from a variety of repositories without encountering cross-origin issues. Fig 7 shows the user interface for toggling the CORS option and instructions for using a third-party CORS service.

The interface for user input of URLs is shown as in Fig 8 . Here we provide users with support for

multiple URL inputs, and users can click the 'Add URL Input' button to add new URLs. At the same time, we also provide support for deleting URLs, and users can delete the input URLs by clicking the Delete button. After accepting the URL input, we will first check the URL. If the user input is empty, an error reminder will pop up and the next step will be paused, as shown in Fig 9.



Fig. 8. URL Input Screen of ATAC Tracks View

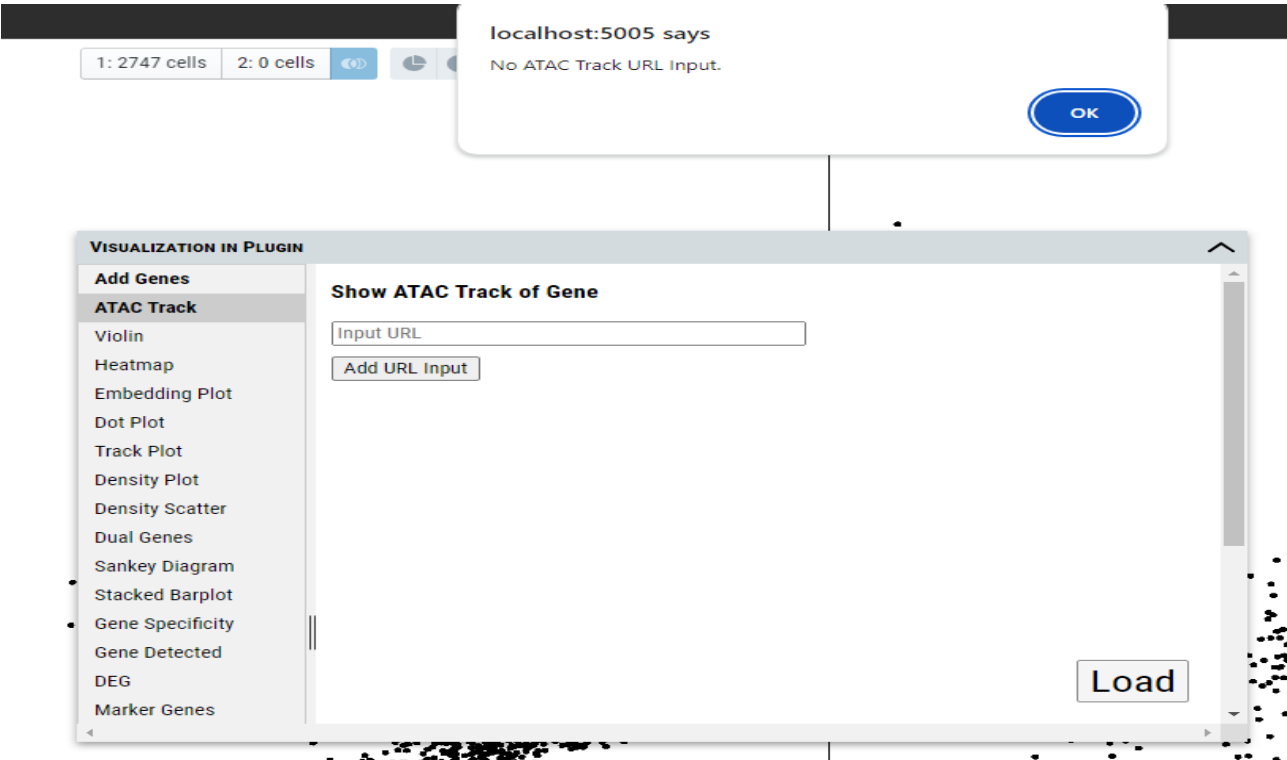


Fig. 9. Error Prompt of ATAC Tracks View

After confirming that the input URLs are correct, we will load IGV.js and show all ATAC Tracks. All the functions supported by IGV.js will also apply here. For example, ATAC Tracks emphasizes interactive exploration, allowing users to focus on specific genomic regions of interest. Users can select

regions within the ATAC track to zoom in and study chromatin accessibility patterns in more detail. As shown in the Fig 10, we display the ATAC Tracks from different sources of the chr3 gene. During the functional testing, we evaluated different capabilities of this implementation, including entering multiple URLs, adding and removing input fields, and handling malformed URLs. Our implementation successfully managed these scenarios and popped up error messages to prompt users to correct any input problems. In addition, the integration of CORS support was verified by enabling and disabling the CORS option, ensuring that data can be accessed and displayed without restrictions in both cases.

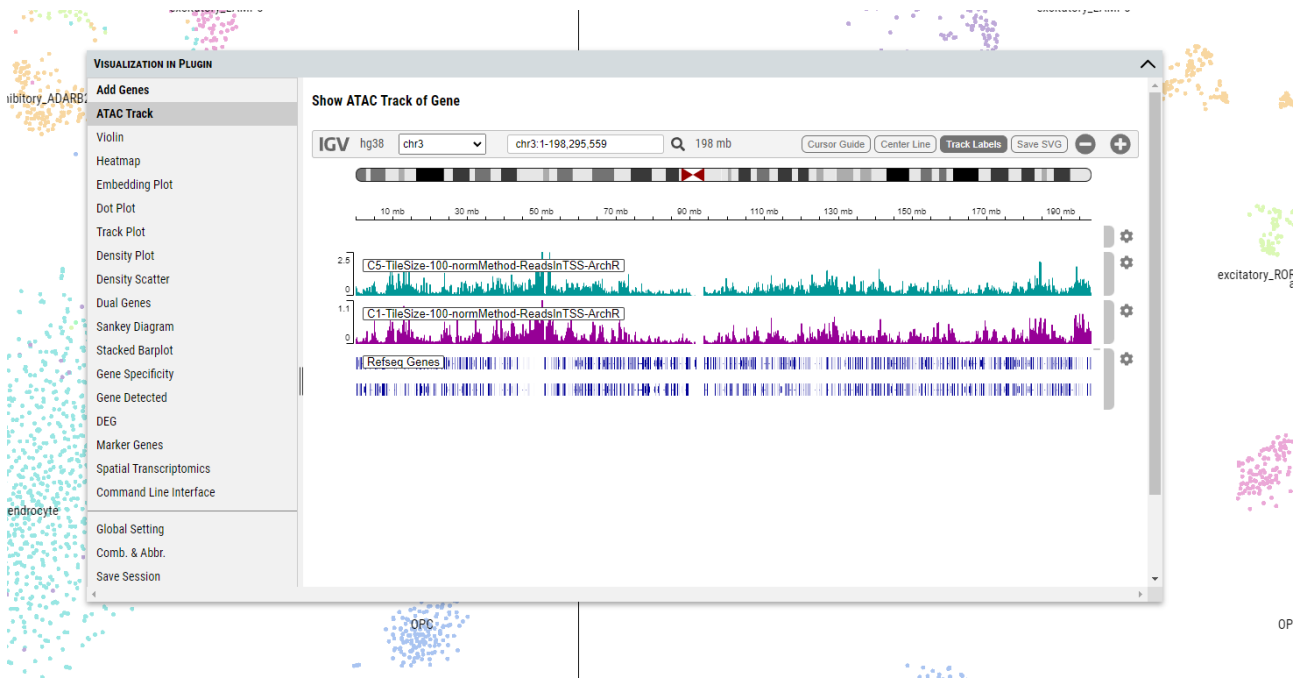


Fig. 10. Tracks Display Screen of ATAC Tracks View

1.2 System Performance

For the implementation of the Cellxgene VIP extension, we performed a performance evaluation to compare the efficiency of using ordinary Anndata data, because we performed server-side data merging and other operations, and the data of the Muon dataset itself is always larger than Anndata dataset due to more information, such as ATAC and GEX in same dataset. This evaluation focuses on the system's loading time, data loading efficiency, and overall stability in different usage scenarios. We perform tests on the time from system loading to page rendering. We used Anndata dataset with 3000 PBMC cells, Muon dataset with 3000 PBMC cells, and dataset with 10000 PBMC cells for loading time evaluation. As shown in the Table 1, the results show that the system will have a significant increase in loading time when using the Muon dataset, and the increase in time is directly related to the size of the Muon data. The time usage for Muon data of the same amount of cells is triple the time needed for Anndata, and when the Muon data gets larger, the longer time are needed to process the internal operations like merge. All the tests are done in same environment, a ThinkPad Laptop with Ryzen 5 Pro 6650U and 16 gigs of RAM. Overall, it is acceptable to use more loading

time because more data needs to be loaded.

Table 1. Time Usage Comparison of Anndata and Muon Data

Dataset Name	Dataset Type	Time Used
3k PBMC	Anndata	12 seconds
3k PBMC	Muon Data	43 seconds
10k PBMC	Muon Data	127 seconds

Throughout the performance evaluation process, our system remained stable, and no major crashes or errors were reported during the test. This stability is critical to ensure that users can rely on the system for complex data analysis tasks without interruption.

1.3 User Case Study

User case studies were conducted to assess whether user interactions with the new modules met usability criteria. We used a variation of the Think Aloud approach, asking participants to record their own thoughts, actions, and questions while completing a series of predefined tasks.

The user case study was done by the project supervisor Dr. Syed Murtuza baker.

Participant was tasked with setting up the application, launching Cellxgene with Muon data, displaying ATAC tracks, displaying two maps on the same screen, and synchronized cluster highlighting. Results showed that the participant was able to successfully complete some of the tasks, with varying amounts of time and errors for different tasks. For example, when setting up the application from GitHub, he encountered difficulties related to environment configuration and dependencies. However, these issues were usually resolved through troubleshooting, and he were able to continue with the tasks.

The task of displaying ATAC tracks was completed successfully. Our tester reported that the CORS functionality was useful, as most university servers block many external accesses, CORS needed to be used in this situation. However, some issues were reported, such as users wanting to be able to download ATAC Tracks, but since IGV.js does not support this feature, we were unable to resolve this issue within this project.

The task of displaying two maps simultaneously and synchronizing cluster highlighting presented additional challenges. Tester found the interface of this feature unintuitive, especially when displaying multiple data views. At the same time, the tester reported the failure of the lasso tool to us, and we immediately investigated and finally determined the cause. During a certain interface update, the lasso tool mistakenly pointed to the right interface where the feature was not available, resulting in this problem. We have resolved this issue in the subsequent new agile development cycle and verified the normal use of the tool. For the synchronization of cluster highlight mapping, we also found problems in expressions. We and the tester had different understandings of the synchronization of cluster highlighting, which made the tester feels that the task could not be done. After sufficient communication with him, we agreed on a consistent explanation and demonstrated the normal operation of the feature.

In addition, the tester reported that our application experience presented a positive experience, and the application startup was very easy, but also indicated some usability issues, such as ATAC Tracks

being displayed in a floating window instead of the main window.

Overall, our application basically met the usability requirements and successfully achieved the project objectives.

1.4 USU Score

The System Usability Scale (SUS) survey was designed to assess the overall usability of our implementation of the Cellxgene VIP extension. Participants rated various aspects of the system on a scale from 1 (strongly disagree) to 5 (strongly agree).

We may consider our system useable and fully functional as it achieved a SUS score of 87.5, above the usual application average score of 68[2]. Additionally, this score shows that although there is certainly space for development, people generally think the system is user-friendly.

Overall, the User Case Study and SUS survey results confirm that the Cellxgene VIP extension is a generally usable system with several strengths, especially in terms of ease of use and user confidence. However, the results also highlight areas for improvement, such as enhancing the integration and intuitiveness of more advanced features.