User Guide

Version: 1.0.0

Prepared by
Chitrasen Mohanty, Lingxin Cheng

Contents:

1. Application overview

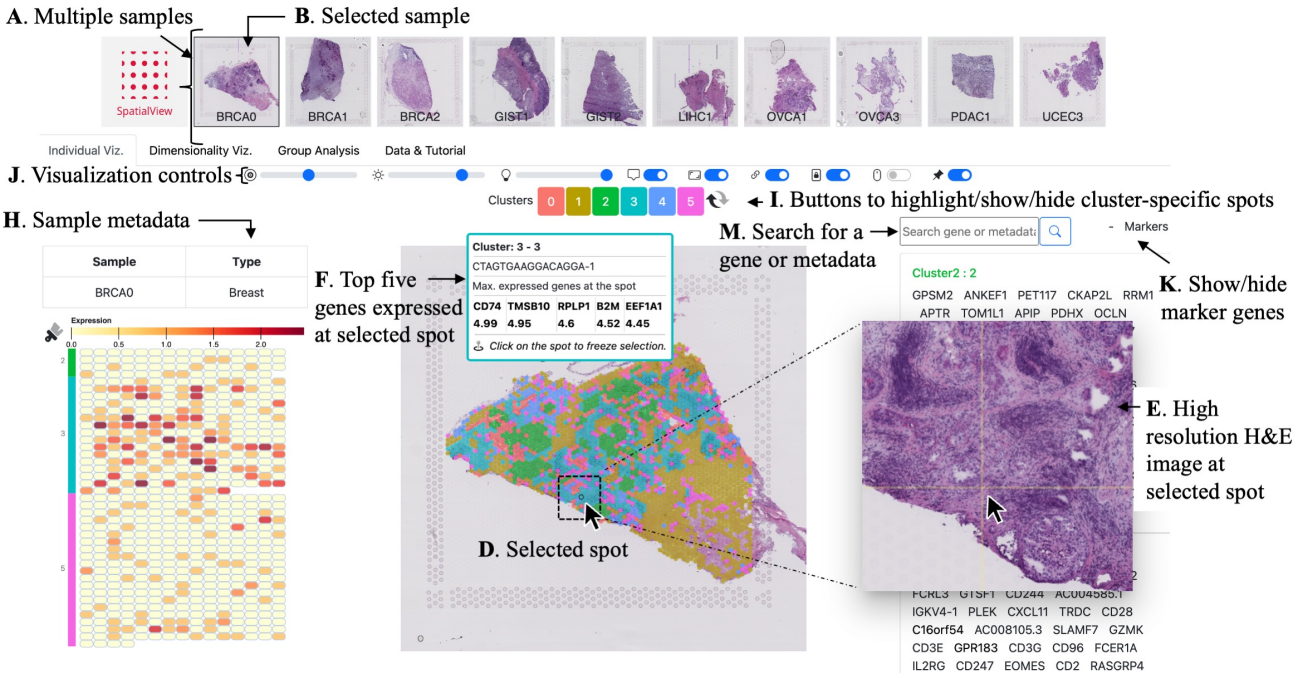
SpatialView is a web browser-based interactive application for visualizing data and results from Spatial Transcriptomic (ST) experiments involving multiple samples. Currently SpatialView is capable of handling data from 10x Visium platform. This is an active project and there are plans to accommodate Slide-seq and other ST platforms.

SpatialView is designed to run all the required computations in the users' browser, thus doesn't require any specialized server and eliminates hassles for server management. When user accesses the application, all the required data are downloaded to the users' browser, and all the required computations for visualizations are done in the browser memory. Because of this design choice SpatialView is faster as compared to Shiny based ST visualization applications.

In the following figures some of SpatialView features are described, however, depicting all the features in text or static figures may not justify all its functionalities. SpatialView is highly interactive, please checkout the demo application to explore the features in real-time:

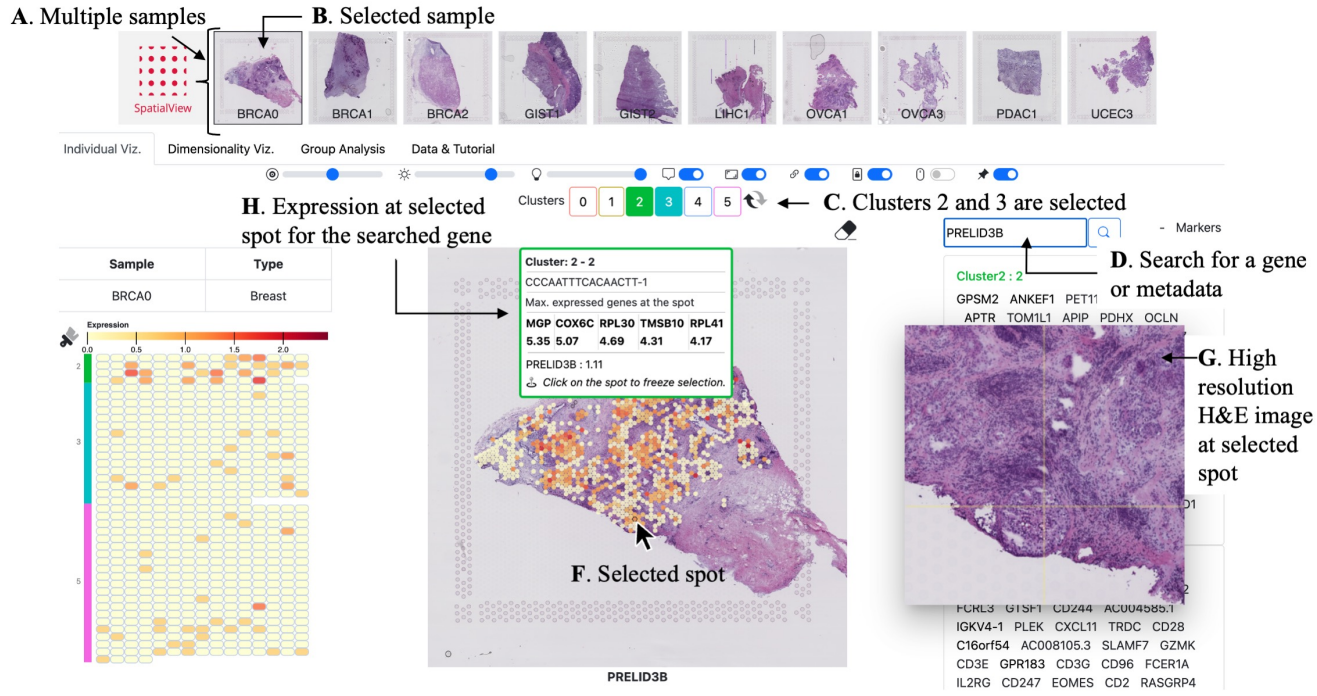
Demo application: https://www.biostat.wisc.edu/~kendzior/spatialviewdemo/

Figure 1:



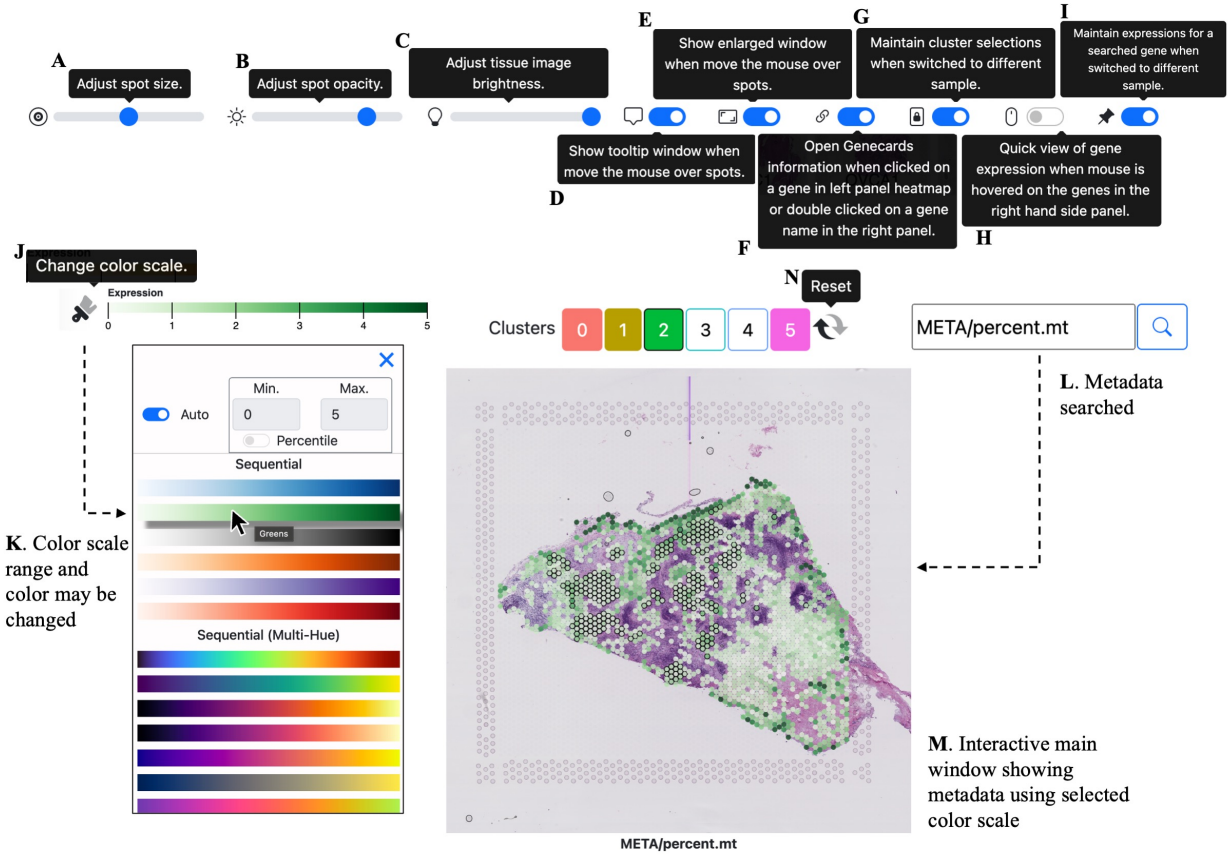
A screen shot of SpatialView (Individual Viz. tab). (A) The top panel of SpatialView shows H&E stain thumbnail images from all samples. (B) For single-sample visualization, a user may click to choose any sample from the top panel. For that sample, an H&E image with spot-specific cluster information is shown (C). Upon mouse over of any spot (D), SpatialView provides a zoomed-in H&E image (E) along with spot-specific information (cluster membership, top five gene expressions) (F) and a heatmap of marker gene expression at that spot (G). Sample metadata such as tissue name and tissue type are also shown (H). A user may choose to highlight, show or hide spots belonging to particular clusters by using the cluster-specific toggle buttons (I). Spot-specific size, opacity, and H&E image brightness can be adjusted using the sliders (J). If marker genes are provided, then they can be listed using the 'markers' link (K) and the cluster specific markers will be visible in the right panel (L). On click of a gene name or by searching a gene (M), the expression of the gene across the spots can be visualized. Documentation about a gene in *genecards.org* can be accessed by a single click on a gene in the heatmap or by double click on a gene name in the list of markers.

Figure 2:



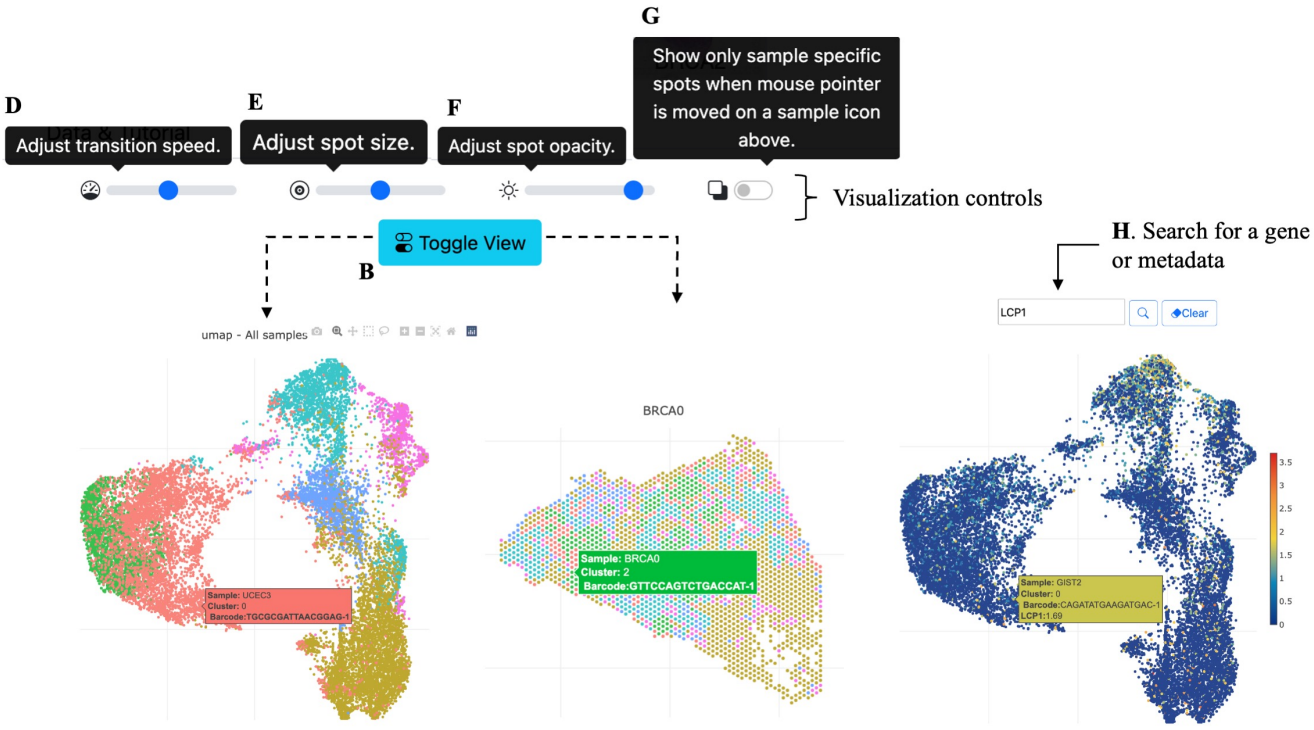
A screen shot of SpatialView (Individual Viz. tab). The top panel of SpatialView shows H&E stained thumbnail images from all samples (A). For single-sample visualization, a user may click to choose any sample from the top panel (B). A user can choose interested cluster specific spots using the toggle buttons (C). After searching a gene name in the search box (D), its expression across the selected cluster specific spots will be shown in the interactive panel (E). On mouse over a spot (F), a zoomed-in H&E image is shown (G); spot-specific information includes expression of the searched gene at the selected spot (H). Marker genes are shown in the right-side panel (I) and heatmap showing expressions of markers at the selected spot is shown in the left-side panel (J).

Figure 3:



Visualization controls (Individual Viz. tab). Sliders for controlling spot size in the main interactive panel (A), spot opacity (B), and Image brightness (C). Toggle buttons for controlling mouseover tooltip (D), zoomed H&E image window (E), genecards pop-up window (F), cluster selection across samples (G), quick view on mouse move over the gene names (H), and retain searched gene/metadata when samples are switched (I). Color scale can be changed using the brush icon (J-K). When metadata is searched in the search bar (L), the interactive window displays the numeric metadata information using the updated color scale. The cluster selections can be reset using the reset button (N).

Figure 4:



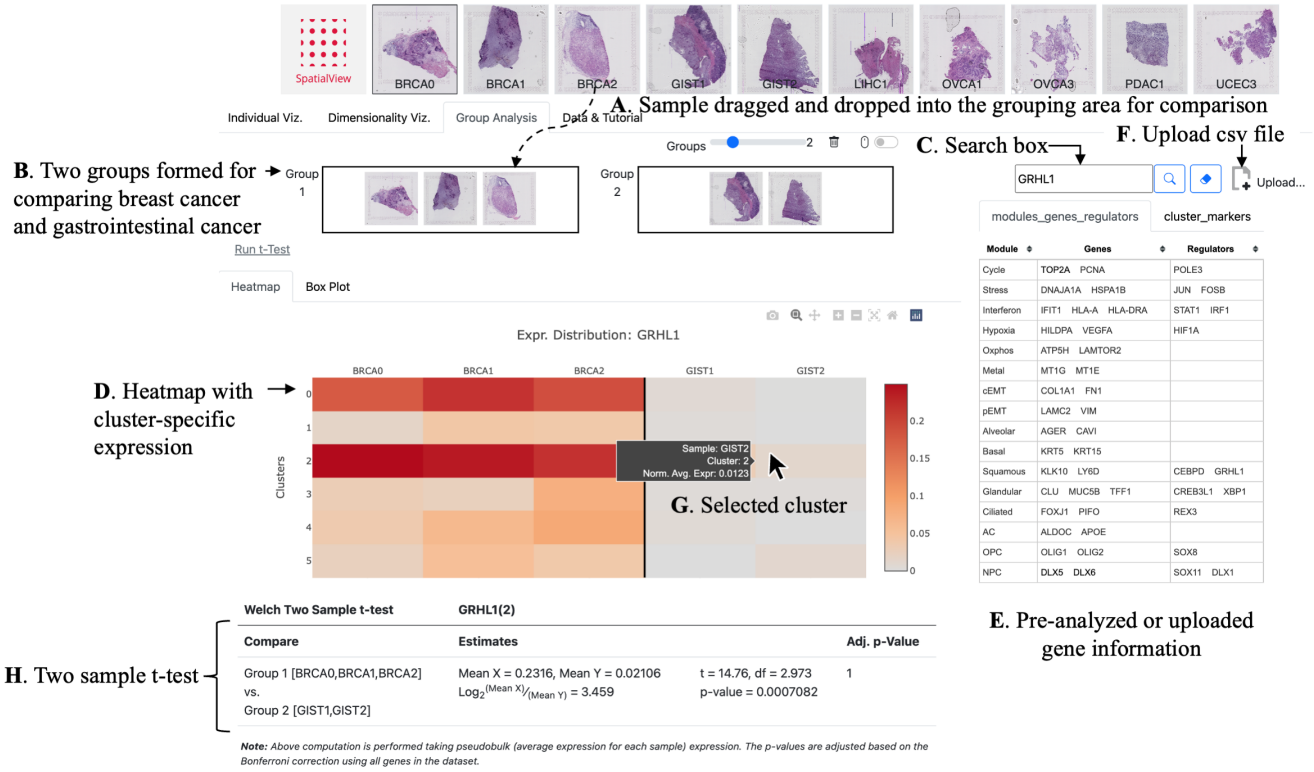
A. UMAP – colored by clusters (spots from all samples)

C. Spatial – colored by cluster (spots from selected sample)

I. UMAP – colored by gene expression (spots from all sample)

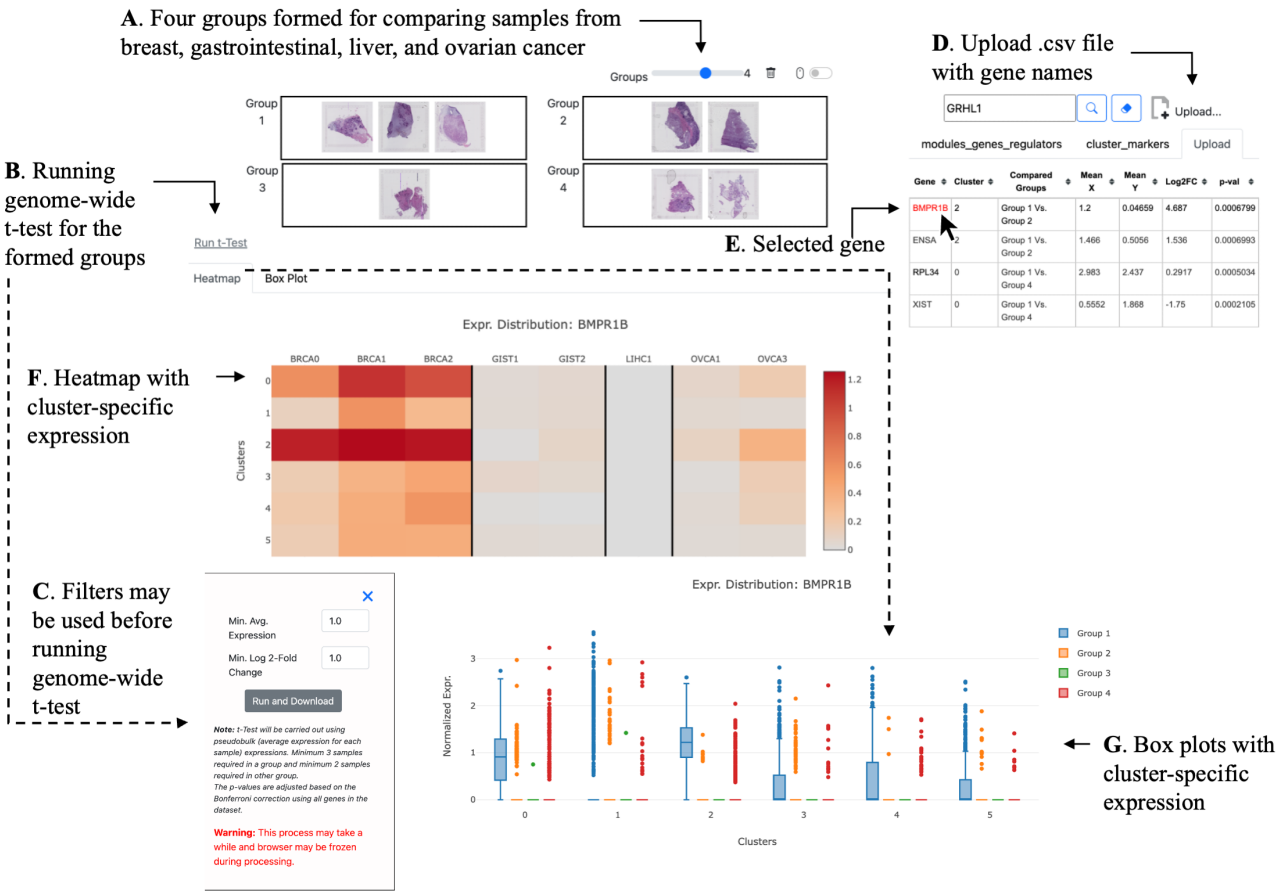
Dimensionality visualization (Dimensionality Viz. tab). 2-dimentional UMAP visualization using all the spots from all samples (A). On clicking the 'Toggle View' button (B), the spots from the selected sample in the top panel (not shown here, same as figure 1B) will be transitioned to corresponding spatial locations (C). The reverse transition i.e. from spatial locations to dimensionality coordinates is done by clicking the 'Toggle View' button again. The speed of the transition may be controlled using the slider (D). The spot size and opacity can be controlled using the slides (E) and (F) respectively. Note that, after changing spot size or spot opacity the figure is reset to UMAP coordinates. While the visualization is rendered as UMAP, mouse hovers on a particular sample in the top panel, the corresponding spots are be highlighted. Only sample specific spots can be visualized by using the toggle button (G). When a gene or metadata is searched in the search box (H), the color of the spots change to represent corresponding expression/feature value (I). User may use additional *Plotly* features (zoom, box select, lasso select etc.) to refine the area of interest.

Figure 5:



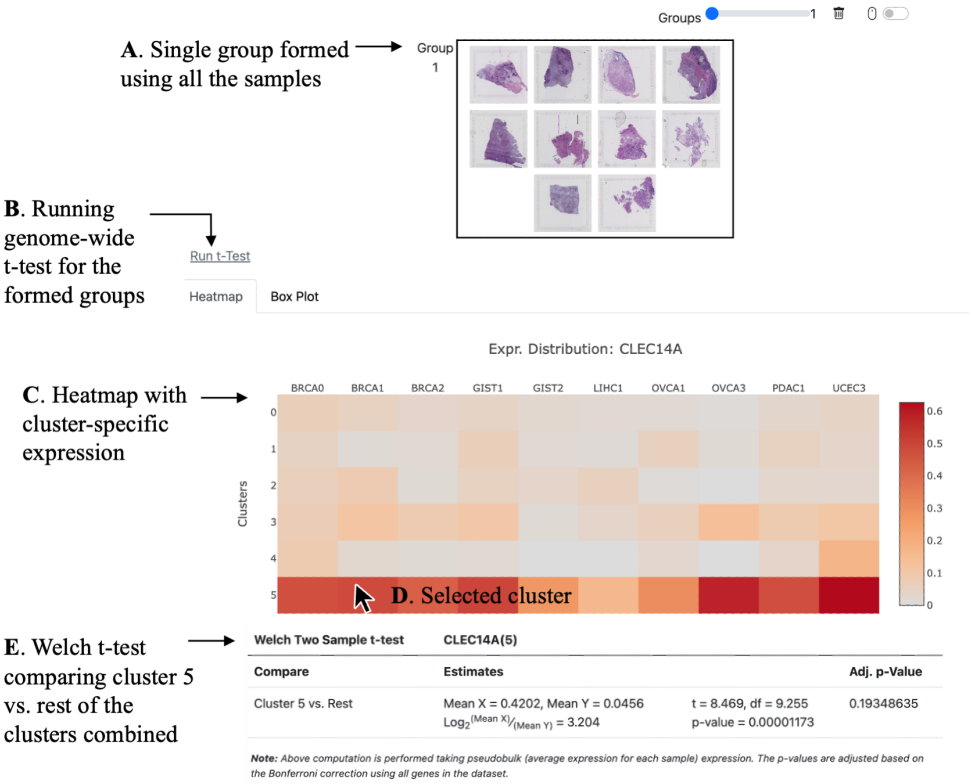
Visualization and analysis of multiple samples (Group Analysis tab). The top panel of SpatialView shows H&E stain thumbnail images from all samples (A). In the 'Group Analysis' tab, samples can be dragged-and-dropped into groups (B). Here, two groups are shown (Supplementary Figure S2 shows visualization with four groups). A user may specify a gene of interest in the search box (C) and cluster-specific expression is displayed as a heatmap (D) or box plots (Figure S2). If a user has conducted an analysis to identify differentially expressed (DE) genes, or some other genes of interest, they can be presented by SpatialView in the right-side panel (E). For example, information from Fig. 1F (Barkley et al., 2022) is shown in the tab 'modules_gene_regulators'. User can also upload (F) any csv file containing gene names and SpatialView will present it as a sortable table (Figure S2). On hovering over the heatmap rows (clusters), details about the sample specific average expression of the selected gene are shown (G), and a two-sample t-test is carried out (provided at least three samples are present in one group and two in another) (H).

Figure 6:



Four-group comparative analysis (Group Analysis tab). Four groups are formed representing four cancer types (breast, gastrointestinal, liver, and ovarian cancer) by dragging and dropping the corresponding samples into the grouping area (A). After forming the groups, a user may search for a gene of interest in the search box to visualize that gene's expression in any cluster, as described in Fig. 2. Additionally, instead of investigating individual genes, a user may initiate a genome-wide pairwise t-test using the 'Run t-Test' link (B). Before running the test, additional filters may be applied to refine the set of genes considered (C). After completion of the tests, a results file will be saved; for visualizing results, the file can be uploaded using the upload link (D). After uploading the results file, an additional tab with a sortable table will appear in the right-side panel. SpatialView automatically detects the gene names from the uploaded file and allows them to be searched interactivity. On mouse over of the *BMPR1B* (E), we see in the table that its expression is higher in group 1 than in group 2 for cluster 2 (log 2-fold change 4.687); cluster wise expressions of *BMPR1B* across the groups is also shown in a heatmap (F) and box plots (G)

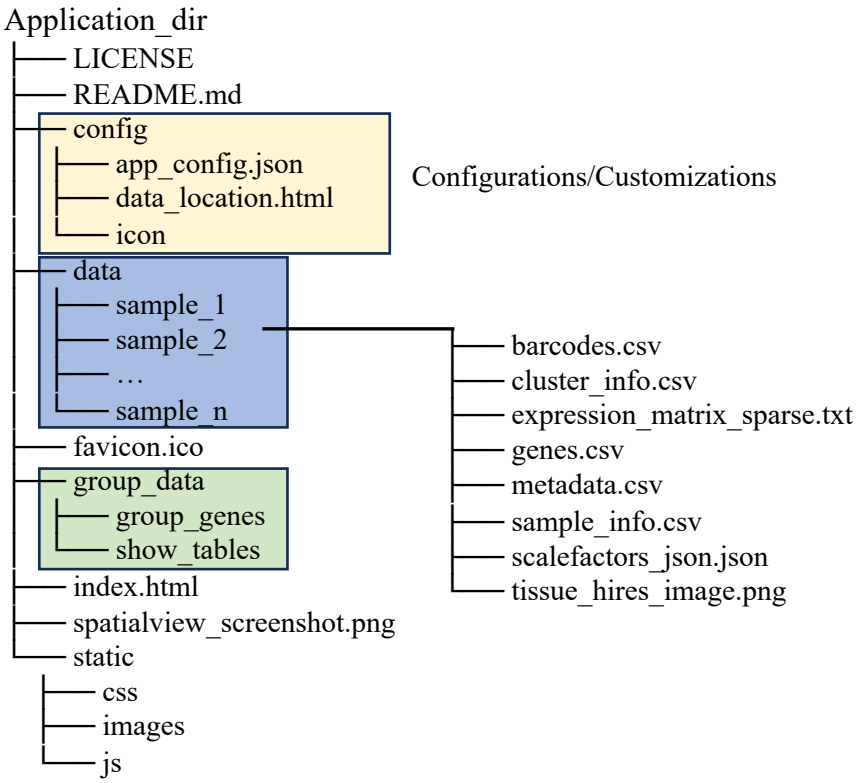
Figure 7:



One-group analysis with comparisons across clusters (Group Analysis tab). For this analysis, a single group is formed and samples of interest are dragged-and-dropped into the grouping the area (A). Once the group is formed, a user may search for a gene of interest (not shown here) or perform a genome-wide search as described in Figure 2 and Figure S2. In the case of one-group, when genome-wide t-tests are carried out using the 'Run t-test' link (B), tests are conducted for one cluster vs. the rest of the clusters. This is also the case on mouse-over of any row (cluster) of the heatmap (C, D).

2. Application structure

Multi-sample support is core to SpatialView. The application is designed to add or remove samples in a plug-and-play concept. As the application is designed to be hosted in any http web server, it follows basic website folder structure. The data from each sample is contained in its independent sub-directory (named after sample name) inside the predefined 'data' directory.



Each sample specific sub-directory contains following required files : barcodes.csv, cluster_info.csv, sample_info.csv, scalefactors_json.json, and tissue_hires_image.png. Additionally, the expression matrix is stored in expression_matrix.csv for dense expression matrix where rows represent features and columns represents spots (barcodes). However, to lower memory usage, SpatialView prefers compressed sparse column-oriented (CSC) format of expression matrix (default). For sparse expression matrix, the expressions are stored in expression_matrix_sparse.txt where each row correspond to (gene_id, spot_id, expression); the gene ids and the corresponding gene names are stored in genes.csv. Similarly, for spots/barcodes information are stored in barcodes.csv file.

Details of the files:

cluster_info.csv: columns are "cluster", "color", "name", "genes" . Example:

cluster	**color**	**name**	**genes**
1	#00BA38	Cluster 1	GPSM2, ANKEF1, PET117, CKAP2L, RRM1
2	#00BFC4	Cluster 2	ADGRE1, SLAMF6, ICOS, IL2RA

metadata.csv: A csv file, each row represents a barcodes. A column containing cluster membership is expected. Example:

barcode	orig.ident	<optional metadata>	seurat_clusters
AAACACCAATAACTGC-1	BRCA0	…	1
AAACAGTGTTCCTGGG-1	BRCA0	…	1

sample_info.csv: A csv file with header and a single row for sample level metadata information. Example:

sample	type
BRCA0	breast

scalefactors_json.json: Scale factor file from Cellranger output.

tissue_hires_image.png: High resolution H&E image from Cellranger output.

Files inside the 'config' directory may be used for customizing/configuring the application. The details are described in the next section.

Group Data

Optionally, additional data with gene names can be to 'group_data/group_genes ' or 'group_data/show_tables ' directories to access them in the 'Group Analysis' tab (Figure 5-E). The files group_data/group_genes required to be same format as cluster_info.csv (previously described). However, files in group_data/show_tables should be in csv format and do not require any additional formatting.

3. Configuration/customization

SpatialView can easily be customized by overwriting the human readable config.json (a key-value paired file) located at
<application_dir>/config/app_config.json

The keys and the values in config.json are self-explanatory. The keys are corresponded to features in the application and the values are default values used. To override the default values, user may directly change the values using any text editor. *These values can also be changed from the R and Python programming environment using SpatialViewR and SpatialViewPy packages respectively (the details are provided in later sections).*

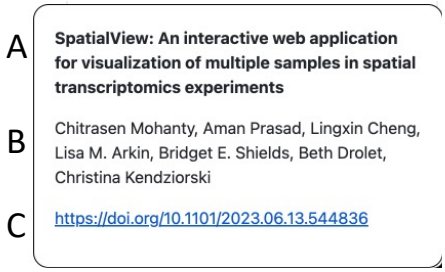
Some of the key features are described here:
a. Toggle controls:

Feature	Usage	Key	Default Value	Options
	Show tooltip window when move the mouse over spots.	tooltip_on	true	[true, false]
	Show enlarged window when move the mouse over spots.	enlarged_window_on	true	[true, false]
	Open *Genecards* information when clicked on a gene in left panel heatmap or double clicked on a gene name in the right panel. The link can be altered using the key 'genecard_url'	genecards_link_on	true	[true, false]
	Maintain cluster selections when switched to different sample.	cluster_lock_on	true	[true, false]
	Quick view of gene expression when mouse is hovered on the genes in the right-side panel.	quick_mouseover_on	false	[true, false]
	Maintain expressions for a searched gene when switched to different sample.	pin_search_on	true	[true, false]

b. Data files

Key	Default value	Description
data_file_name_expressions	expression_matrix.csv	‘expression_matrix.csv’ should contain the gene expressions. SpatialView prefers sparse encoding and looks for expression_matrix.csv if ‘expression_matrix_sparse.txt’ is not located.
data_file_name_expressions_sparse	expression_matrix_sparse.txt	Expressions in sparse column0oriented (CSC) format.
data_file_name_genes	genes.csv	Gene names with gene names in a row.
data_file_name_barcodes	barcodes.csv	Barcodes with each barcode in a row.
data_file_name_cluster_info	cluster_info.csv	csv file with clusters and associated genes. Each row represents a row.
data_file_name_metadata	metadata.csv	csv file with meta data information, analogs to Seurat metadata.
data_file_name_scalefactor	scalefactors_json.json	A json file output from 10x Spaceranger.
data_file_name_sample_info	sample_info.csv	csv file for sample metadata information. Each row represents for a sample.
image_file_name_high_resolution	tissue_hires_image.png	High resolution image file that SpatiaView uses.
data_cluster_column	seurat_clusters	Column name in the metadata csv file that contains cluster names for each barcode.

b. Citation



Feature	Key	Default value	options
A	cite_title	SpatialView: An interactive web application for visualization of multiple samples in spatial transcriptomics experiments	String: Title of the paper
B	cite_authors	Chitrasen Mohanty, Aman Prasad, Lingxin Cheng, Lisa M. Arkin, Bridget E. Shields, Beth Drolet, Christina Kendziorski	String : Author names
C	cite_link	https://doi.org/10.1101/2023.06.13.544836	URL: doi link

d. Dimensionality Visualization

Key	Default value	Description
dim_plot	umap	A valid reduced dimension feature. SpatialView uses only first 2-dimentions to visualize the dimensionality plot. The first 2 coordinates are stored in metadata.csv file. For example, the umap coordinates are stored in umap_1 and umap_2 columns. Other choices may be ‘tsne’ or ‘pca’

To change the logo of the application, icon.png file may be replaced in <Application_dir>/config/icon. Note that the image file name and size should not be altered.

The details of the data information shown in the ‘Data and Tutorial’ tab may be updated in the <Application_dir>/config/data_location.html file.

4. Running SpatialView

Spatial Transciptomic (ST) Visium data from 10x experiments can be visualized in **SpatialView** multiple ways.

A. Using R

- To run SpatialView from R environment you may use *SpatialViewR* package.

Currently *SpatialViewR* supports *Seurat* and *SpatialExperiment*. Please check the step-by-step tutorials for details.

A step by step guide to export data from *Seurat* object

A step by step guide to export data from *SpatialExperiment* object

By setting 'downloadRepo = TRUE' (default) in the SpatialViewR functions, latest SpatialView files are downloaded from its GitHub repository in the background. If downloadRepo is set to FALSE, then it's required to download the SpatialView manually and set the 'exportPath' to the data directory (see Application Structure section).

Note that, when downloadRepo is set to False, launchApp option is ignored.

B. Using Python

- To run SpatialView from Python environment you may use *SpatialViewPy* package.

Currently *SpatialViewPy* supports *Scanpy*.

A step by step guide to export data from *Scanpy* object

By setting 'downloadRepo = True' (default) in the SpatialViewPy functions, latest SpatialView files are downloaded from its GitHub repository in the background. If downloadRepo is set to False, then it's required to download the SpatialView manually and set the 'exportPath' to the data directory (see Application Structure section).

Note that, when downloadRepo is set to False, launchApp option is ignored.

C. Using code from GitHub

The latest SpatialView application can be downloaded from GitHub (*spatialview-latest* release) and can be run in local machine by following steps.

Note that, application can run from any http server (web server), however the following steps assume that Python is installed on the local machine and the application runs in Python http.server.

Download the file from here (*spatialview-latest* release is recommended) to your local system and unzip the folder. Your processed data to be placed in the data directory inside the unzipped SpatialView directory.

Each sample should have its own directory and may contain following files for a sample:

1. Expression matrix:

Option 1 - *Sparse matrix* (preferred): compressed sparse column-oriented (CSC) format, barcodes.csv, genes.csv

```
#expression matrix
normalized_counts <-    as(SummarizedExperiment::assay(speObj, i = 'assayName'), "dgCMatrix")
normalized_counts <- round(normalized_counts, 2) # round the expressions to reduce space
Matrix::writeMM( normalized_counts,  file = "<DATA/SAMPLE_NAME>/expression_matrix_sparse.txt ")
# barcodes
write.csv(colnames(normalized_counts), ),  "<data/SAMPLE_NAME>/ barcodes.csv",
                                        row.names = FALSE,  quote = FALSE)
 # genes
 write.csv(rownames(normalized_counts),  "<data/SAMPLE_NAME>/ genes.csv",
                                        row.names = FALSE, quote = FALSE)
```

Option 2 - *Dense matrix:* a csv file with barcodes as columns and genes name in an additional column

```
write.csv(normalized_counts, , "<data/SAMPLE_NAME>/ expression_matrix.csv ",
                row.names = FALSE, quote = TRUE)
```

5. Publishing data/hosting application

To host the application, a user needs to copy the output directory (see 'Application structure' section) to a server which can access web request. Yes, it's as easy as it sounds.

Note that user may like to change the details related to the sample data in the <Application_dir>/config/data_location.html file (see ''Configuration/customization' section).

6. Related URLs

Here are some useful URLs related to SpatialView

Description	URL
Demo application	https://www.biostat.wisc.edu/~kendzior/spatialviewdemo/
SpatialView page	https://kendziorski-lab.github.io/projects/spatialview/spatialview.html
SpatialView Github	https://github.com/kendziorski-lab/spatialview
SpatialViewR Github	https://github.com/kendziorski-lab/SpatialViewR
SpatialViewPy Github	https://github.com/kendziorski-lab/SpatialViewPy