

SPATIAL TRANSCRIPTOMICS VIEWER v0.6.3 USER MANUAL

Jose Fernandez Navarro

November 14, 2016

Preface

The Spatial Transcriptomics viewer (ST viewer) is a desktop application that allows users to securely access and visualize spatially distributed gene expression profile data with their respective tissue image. At the same time it allows users to analyze the data directly, or export it for their own analyses. The application can obtain the data from a secured server (configuration files will need to be updated for this) and/or from local files.

For installation instructions, please see the README.md file. This file describes the install instructions on Windows, OSX and Linux.

Licensing

This is open source software under the MIT license. Read LICENSE for more information about the licensing terms.

Contact

Please report bugs, feedback, questions and errors to:

Jose Fernandez Navarro (jose.fernandez.navarro@scilifelab.se) or

Alexander Stuckey (alexander.stuckey@scilifelab.se)

Contents

1 Spatial Transcriptomics Technology	3
1.1 What is Spatial Transcriptomics	3
1.2 What is a Spatial Transcriptomics dataset	3
2 Installing and Running the Spatial Transcriptomics Viewer	5
2.1 Installing the viewer	5
2.2 Running the Viewer	5
3 General Overview	7
3.1 Online Database	7
3.2 User Interface	7
3.3 Gene Display Modes	10
3.4 Importing a local dataset	12
4 Using the Viewer	15
4.1 Selection Mode	15
4.2 Differential Expression Window	17
5 Known Issues and Future Plans	19
5.1 Known Issues	19
Appendices	21
Appendix A	23
Installing in OSX	23
Installing in UNIX	24
Installing in Windows	25

List of Figures

3.1	The spatial transcriptomics viewer default interface	7
3.2	Toolbar showing username after logging in.	8
3.3	Datasets view.	8
3.4	Default view after loading a dataset	9
3.5	Gene List	9
3.6	The toolbar when a dataset is loaded.	10
3.7	The visual canvas of the ST viewer in each map mode.	10
3.8	Gene display configuration	11
3.9	The local dataset import window.	12
4.1	The visual canvas of the ST viewer showing different selection types.	15
4.2	The selection window.	16
4.3	The genes in the current selection.	16
4.4	The differential expression window.	17

Chapter 1

Spatial Transcriptomics Technology

1.1 What is Spatial Transcriptomics

Spatial Transcriptomics is a method that allows visualization and quantitative analysis of the transcriptome in individual tissue sections. By placing histological sections on glass slides with arrayed oligonucleotides containing positional barcodes, it is possible to generate high quality cDNA libraries with precise positional information for RNA-seq. This provides transcriptomes data in a versatile format for bioinformatic analyses of gene expression within the tissue context, valuable on both research and diagnostics.

1.2 What is a Spatial Transcriptomics dataset

A Spatial Transcriptomics dataset is a tab separated text file containing counts for each gene at each position on the array (table 1.1).

Table 1.1: A general example of a Spatial Transcriptomics dataset file.

	Gene A	Gene B	Gene C	...
1x1	12	0	27	...
2x1	64	62	63	...
3x1	375	43	0	...
...

This dataset is generated from the ST_Pipeline (obtainable from GitHub, https://github.com/SpatialTranscriptomicsResearch/st_pipeline). However, although the ST_Pipeline outputs a tab separated file, the current version of the ST_viewer (v0.6.3) only supports data in the JSON file format. To convert from .tsv files to .json files, the python script matrix_to_json.py can be used. It can also be obtained from GitHub (https://github.com/SpatialTranscriptomicsResearch/st_analysis), in the scripts folder. Further releases of the viewer will be able to read .tsv files directly, eliminating the need to convert to .json first.

Chapter 2

Installing and Running the Spatial Transcriptomics Viewer

2.1 Installing the viewer

The viewer can be installed for Windows, MacOSX and UNIX PCs. Build instructions can be found on the spatial transcriptomics [GitHub page](#). The same instructions can also be found in the README.md file in the GitHub repository. They can also be found in Appendix 5.1 in this manual.

2.2 Running the Viewer

MacOSX

To run the viewer on OSX, simply double click the app icon that you built in the previous step. It can be found in /path/to/build. For ease of use, the app can be copied to the applications folder. It can then be opened using spotlight search (press command + space, then type stviewer).

Windows

To run the viewer in Windows, double click the desktop icon (if created during install), click the start menu shortcut (Start → Programs → STViewer, if created during install) or navigate to the folder where the viewer was installed and double click the icon.

UNIX

To run the viewer in UNIX, open a terminal and type /path/to/bin/STViewer. Alternatively, you can add the path to the binary to the system PATH variable, create a symlink to the binary in /usr/bin or \$HOME/bin (if it exists and is in your \$PATH) or move the binary to /usr/bin or /\$HOME/bin. To add a directory to your \$PATH variable, open the .bashrc file (or equivalent e.g. .zshrc) found in your \$HOME directory with your favourite text editor. At the end of the file add the following line: `export PATH="$PATH:/path/to/directory/"`. Then, close your terminal or enter source .bashrc (or equivalent). The .bashrc file is a hidden file, so it is not normally visible in the directory. The easiest way to edit the file is to open a terminal and type gedit .bashrc.

Chapter 3

General Overview

3.1 Online Database

The online mode is only supported if the user sets up their own ST server and configures the database and ST API/ADMIN correctly. Instructions for how to do this can be found on GitHub at the following address https://github.com/SpatialTranscriptomicsResearch/st_api. How to use the viewer in offline mode can be found in chapter 3.4.

3.2 User Interface

When running the spatial transcriptomics viewer for the first time, you will be presented with an interface as shown in figure 3.1. If the viewer has been configured to access datasets through the ST API then the login window will be shown, otherwise it will not be shown.

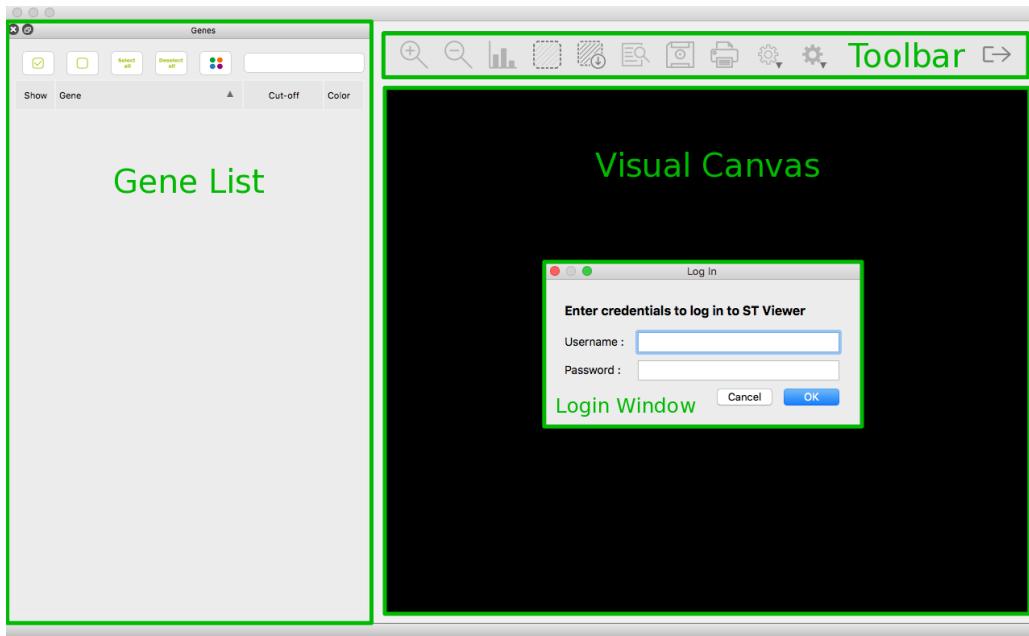


Figure 3.1: The default interface presented when the ST viewer is run for the first time. The elements present are the gene list window on the left, the toolbar in the top right, the visual canvas on the right, and optionally the log in window (if the viewer is configured to access datasets through the ST API).

The menu bar has two options, `File` and `Views`. In the `File` menu you are able to view information about the viewer, as well as clear the local cache (doing so will force any datasets to be re-downloaded or re-imported). This is useful when changes have been made to the external datasets. The `Views` menu toggles the display of the datasets, selections and genes windows. It also contains the logout toggle, if you can connect to an online database.

Once you have logged in (if applicable), your user name will be displayed in the toolbar, as shown in figure 3.2.



Figure 3.2: Toolbar showing username after logging in.

There are two other main windows in the ST viewer, datasets and selections. They can both be accessed from the views menu by clicking **views → datasets** and **views → selections**. The datasets window shows all the datasets that you have access to. There is a search box that can be used to find a keyword in the dataset names to narrow down the list of datasets (see figure 3.3). There are five buttons to the right of the dataset search box (labelled 1 — 5). They are:

1. Import dataset from file (local file on your computer)
2. Open selected dataset (for datasets stored in the database)
3. Edit selected dataset (change name, add comments)
4. Delete selected dataset
5. Refresh list of datasets

The datasets displayed in the dataset view can also be sorted by each column (e.g. name or species).

 A screenshot of the 'Datasets' view. At the top, there is a search bar containing the text 'test'. Below the search bar is a table with columns: Name, Tissue, Species, Created, and Last Modified. The table lists several datasets, including '141031_AL_test_C1', '141031_AL_test_C1.ny', '150127_1000L2_CN95_C1_PTSB_150203_new_test_SG', '150819_1000L2_CN12_D1_PTWBinsol_merged_pip075_test', '160413_Mickan_test', 'test', 'test2', 'Test_HC_C1', and 'Test_HC_C2'. The 'Created' and 'Last Modified' columns show dates and times. To the right of the table are five numbered buttons labeled 1 through 5, each with a corresponding icon above it.

Name	Tissue	Species	Created	Last Modified
141031_AL_test_C1	Xenograft	Homo sapiens/Mus ...	10/31/14 2:36 PM	10/31/14 3:29 PM
141031_AL_test_C1.ny	Xenograft	Homo sapiens/Mus ...	10/31/14 3:42 PM	10/31/14 3:42 PM
150127_1000L2_CN95_C1_PTSB_150203_new_test_SG	Leaf bud	Populus tremula	2/9/15 2:44 PM	5/18/16 10:26 AM
150819_1000L2_CN12_D1_PTWBinsol_merged_pip075_test	Leaf bud in solution	Populus tremula	5/18/16 11:21 AM	5/18/16 11:21 AM
160413_Mickan_test	Mickan_test	Mickan_test	4/13/16 5:15 PM	4/13/16 5:15 PM
test	test	test	6/8/16 5:34 PM	6/8/16 5:34 PM
test2	test	test	6/8/16 5:44 PM	6/8/16 5:44 PM
Test_HC_C1	Hippo	Human	2/25/16 5:11 PM	4/18/16 12:54 PM
Test_HC_C2	Hippo	Human	2/25/16 5:57 PM	2/25/16 5:57 PM

Figure 3.3: Datasets view. The search box is being used to display only the datasets that contain the word "test" in their name.

The selection view will be discussed further in chapter 4.

Note: Each dataset is associated with a substantial amount of data, which will be accessed after a dataset has been selected and opened. As such the transition between the views might require a few moments. It is worth noting that the data is cached, which implies that once it is downloaded, the next time the same dataset will open much quicker. This does not apply if the dataset is updated in the cloud in the meantime.

Opening a new dataset while a previous dataset is open can be done in the same manner as opening a dataset for the first time. Note that when this is done the visualisation settings are set back to the defaults.

When the dataset is loaded, the visual canvas will display with a picture of the tissue. This picture can be zoomed (using the scroll-wheel on the mouse) and panned by clicking and holding left click while dragging the mouse around.

When a dataset is loaded, the gene list will populate with the list of all genes that are associated with the current dataset, and the visual canvas will display the image of the tissue section (see figure 3.4).

The gene list is headed by five buttons and a text box. The buttons are, in order:

1. Show selected
2. Hide selected
3. Select all
4. Deselect all
5. Set colour
6. Search box

The text box enables you to search the list of genes for genes that are of interest. Shown genes are marked with a checkmark in the box to the left of the gene name / ensemble ID. Selected genes have their names highlighted in orange instead of green, and have a darker grey background. Note that shown genes and selected genes may differ (as seen in figure 3.5). The colour column allows you to choose custom colours for single genes or groups of genes, in order to increase visual clarity. The gene list can also be sorted by gene name and cut-off value.

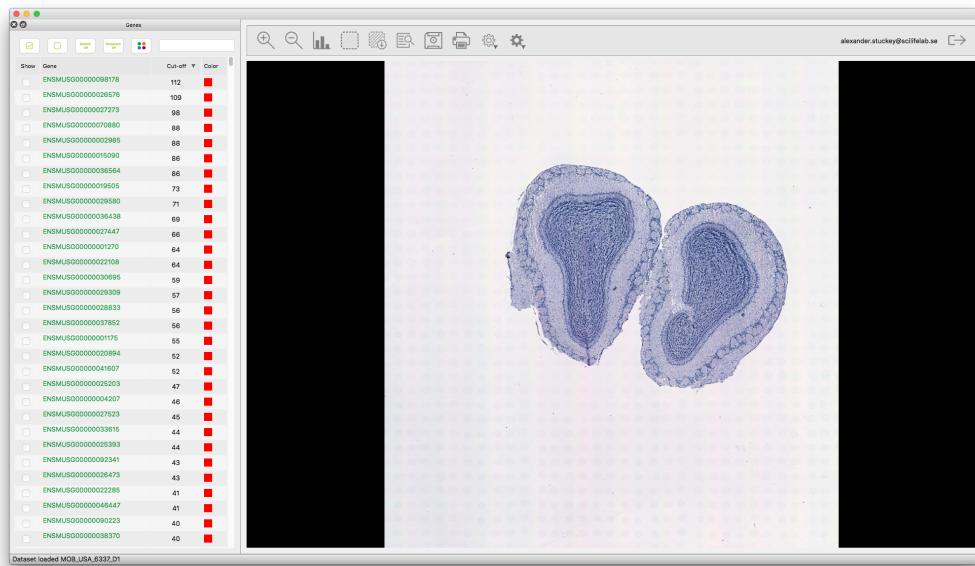


Figure 3.4: The default view after loading a dataset. The gene list is populated by all the genes in the data, and an image of the tissue section is presented on the visual canvas.

The cut-off value is only important (and only makes sense) if all of the positions on the array are being used. If the st_data file has been edited so that only the positions that are under the tissue are being used, the cut-off should be disregarded. Additionally, the option Individual gene cut-off in the Gene display configuration menu should be disabled.

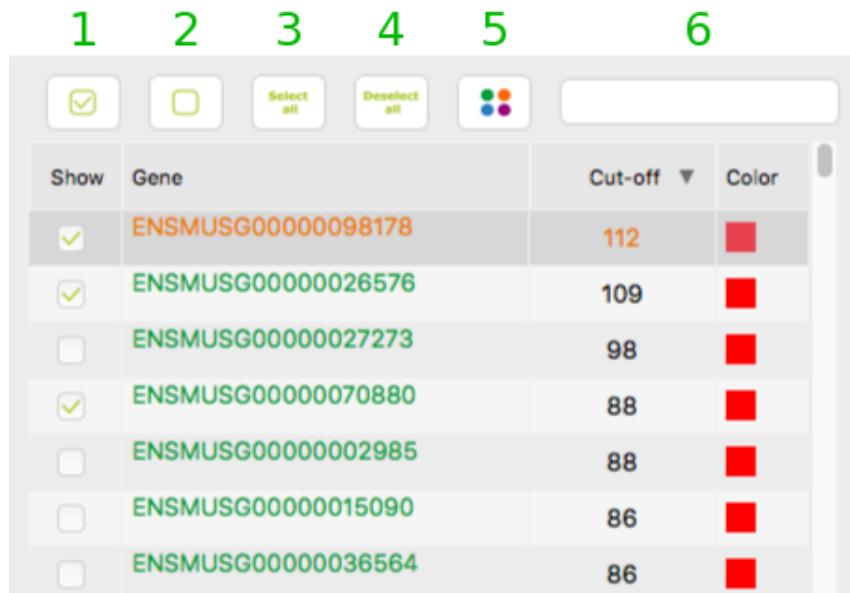


Figure 3.5: Gene list for the current dataset, sorted by cut-off value.

When a dataset is loaded into the viewer the buttons above the visual canvas become active (figure 3.6). The buttons on the toolbar are:

1. Zoom in
2. Zoom out
3. Feature distribution histograms
4. Selection mode toggle
5. Create selection object
6. Feature selection by regular expression
7. Save a picture of the visual canvas
8. Print a picture of the visual canvas
9. Gene display configuration

10. Visual canvas options
11. Logout button



Figure 3.6: The toolbar when a dataset is loaded.

The visual canvas option menu allows us to display a legend for the overall expression values for each spot (useful in heatmap mode). It can also be used to toggle display of the tissue image (and any secondary image) off and on again.

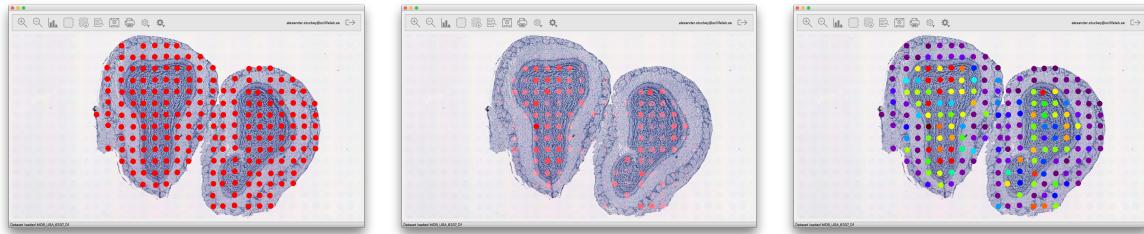
The gene display configuration is shown in figure 3.8. From top to bottom, the options are:

Show Grid toggles the display of the original array and its border simplified to 5×5 features per square.

Show Genes toggles the display of the currently selected genes.

Choose Colour Grid opens a menu to change the colour of the displayed array grid.

3.3 Gene Display Modes



(a) Visual canvas in normal mode. (b) Visual canvas in dynamic mode. (c) Visual canvas in heatmap mode

Figure 3.7: The visual canvas of the ST viewer in the normal, dynamic and heatmap map modes. All genes are selected in these images, with all colours set to the defaults.

In **Normal Mode** (figure 3.7a) the selected genes in each feature are treated independently. When multiple genes are selected, the final colour of the feature will be a mix of the different colours of the selected genes. In normal mode, the threshold setting treats the genes in each feature separately. Therefore, the threshold can hide some of the selected genes in a feature, or even all the genes in a feature. In all cases, the final colour of the feature will be adjusted accordingly.

In **Dynamic Range Mode** (figure 3.7b) the selected genes in a feature are all treated as one unit (as opposed to normal mode). This means that each feature is treated as a single unit where the level of expression is determined by the combination of all the genes selected in the feature. The final expression is normalised and the alpha channel (the brightness) of the feature is adjusted according to the normalised final expression of the feature. This way, users can get an idea of the total expression level of a feature compared to all other features. The colour of each feature will be computed in the same way as in the normal mode (it combines the colours of the genes in each feature). The threshold in dynamic mode affects features as a whole, instead of gene-by-gene as in normal mode. When the threshold is changed the intensity for all the features is re-calculated, as the normalisation factor might have changed.

Heatmap Mode (figure 3.7c) treats genes in a feature in the same manner as the dynamic range mode. Instead of adjusting the brightness of the colour, a new colour is computed for the feature representing the overall level of expression. The low frequencies are blue and represent a low hit count, while the high frequencies are red and represent a high hit count. The heatmap spectrum and threshold levels can be seen on the legend, when the legend is enabled. The threshold works the same way as in dynamic range mode, when the threshold is changed the colour range and colour of features will be recalculated.

The individual gene cut-off is used in cases where the ST_data file contains information for every spot, not just the spots that are under the tissue. In this case, the viewer calculates a cut-off for each gene where expression values under the cut-off are assumed to be outside of the tissue and removed from the display. Expression values that are higher than the cut-off value are assumed to be under the tissue and retained. Leaving this setting on when there is only data for spots under the tissue can result in odd behaviour, and masking of much informative data. It is recommended to disable this setting when the dataset has been edited to remove all spots that are not under the tissue.

The opacity, size and shape parameters change the display of the features on the visual map mode.

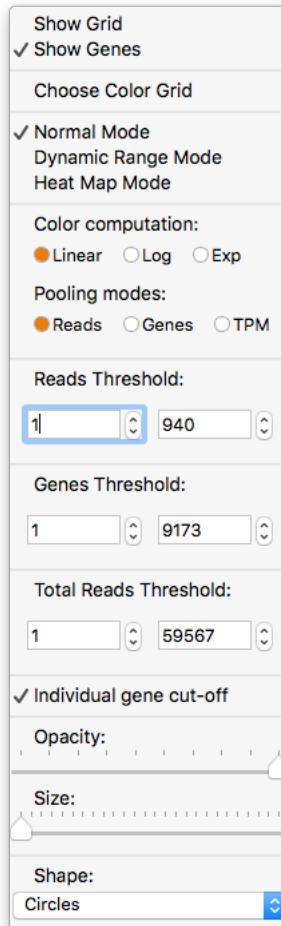


Figure 3.8: Gene display configuration

3.4 Importing a local dataset

To import a dataset locally, click Views → Datasets, then click Import Dataset. This brings up the import window (figure 3.9).

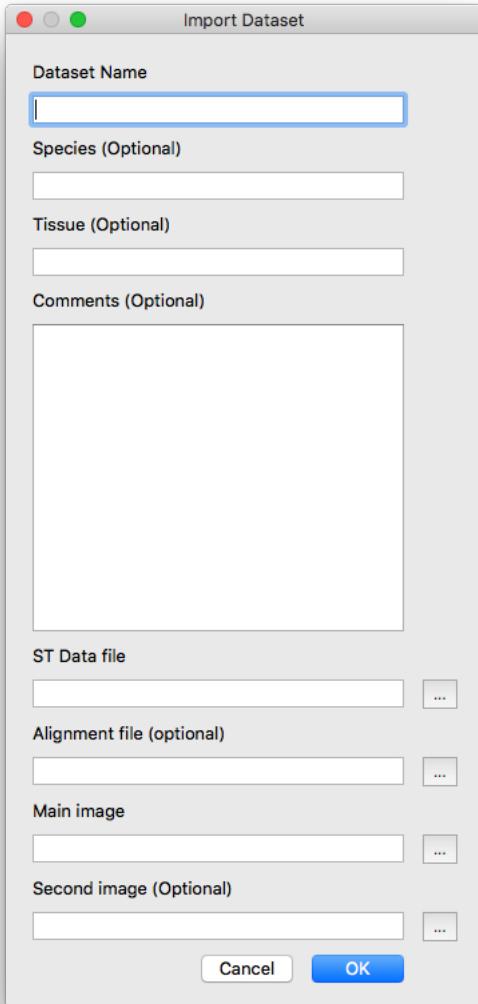


Figure 3.9: The local dataset import window.

To import a dataset, you need to supply the information for each field. It is recommended to fill in each field (except for the second image field) so that it is easy to see what the dataset contains. The ST data file is the .json file obtained after running the st pipeline and converting the resulting .tsv file to .json format using the python script `matrix_to_json.py` described in chapter 1. The main image is the H&E stained image of the tissue.

The alignment file is used to convert between the array coordinates in the st.data file (e.g. 1x2, 23x13) to exact pixel coordinates in the H&E stained image. Doing so allows spots to be displayed in the correct position in the tissue image. The alignment matrix is in the following format:

Table 3.1: General alignment matrix format.

$$\begin{array}{ccc} \text{scaling x} & 0 & \text{offset x} \\ 0 & \text{scaling y} & \text{offset y} \\ 0 & 0 & 1 \end{array}$$

The scaling factor for x is the width of the image (in pixels) divided by the number of spots in one row (32). The scaling factor for y is the height of the image (in pixels) divided by the number of spots in one column (34). These numbers should be entered to one decimal place. An example off an scaling for x and y would be 294.2, 296.3. The offset describes how

far from the top left corner of the image the center of the first spot is located. This value is the same as the scaling values, but negative. Examples of the offset using the same scaling values as before would be -294.2, -296.3. This means that we need to move 294.2 pixels right and 296.3 pixels down from the top left corner of the image to find the center of the first spot.

The alignment matrix that is needed for the viewer is a single tab seperated line of nine values. These can be obtained from the alignment matrix previously calculated. It has the following format:

Table 3.2: General alignment file form for the ST viewer.

scaling x	0	0	0	scaling y	0	offset x	offset y	1
-----------	---	---	---	-----------	---	----------	----------	---

Using the example numbers from earlier, the final alignment matrix example would be:

Table 3.3: An example of an alignment file for the ST viewer.

294.2	0	0	0	296.3	0	-294.2	-296.3	1
-------	---	---	---	-------	---	--------	--------	---

It is important that the alignment file is written in a plain text editor and saved as a `.txt` file. Other editors and other file types can often insert invisible meta-characters into the file that will cause the viewer to crash. Nano, a terminal based text editor available on most Unix systems, is a good choice for writing alignment files. On Windows, notepad or notepad++ would be good alternatives.

All the different views are undockable from the main ST viewer window and can be shown at the same time.

Chapter 4

Using the Viewer

4.1 Selection Mode

Selection mode is toggled using button 4 in 3.6. When selection mode is enabled the user is able to select features on the canvas by clicking and dragging a box on the visual canvas (figure 4.1a). The user can select disjointed areas by holding down shift while selecting an area (figure 4.1b). Areas can also be removed from the selection by either shift + control (Windows and Linux) or shift + command on Mac (figure 4.1c).

In addition to selecting via clicking and dragging on the visual canvas, users can also select genes via regular expressions (button 6 in figure 3.6). Regular expressions provide the user with a powerful means to select specific genes by matching the gene names against a specific string pattern. The regular expression syntax is modeled after the Perl regular expressions. In addition the selection dialog provides three options; to include ambiguous genes, to select non-visible genes, and to make the regular expression case sensitive. By default the regular expression selection excludes ambiguous genes and ignores case differences.

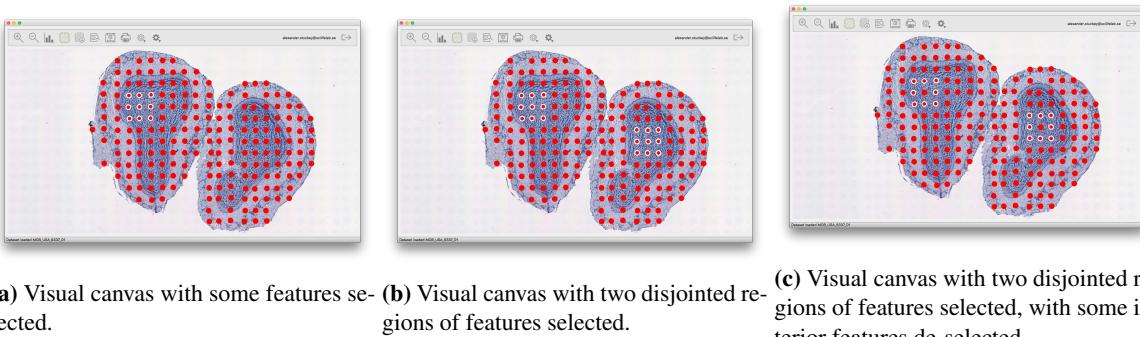


Figure 4.1: The visual canvas of the ST viewer showing the different ways of selecting features.

When the desired features are selected, the user can create a selection object by clicking button 5 in 3.6. The selections can then be viewed by clicking on views → selection. This brings up the selection window (figure 4.2).

The selection menu lists all the selections that have been made in the current sample. The textbox on the top left enables the user to narrow down the list of selections. The buttons on the right, in order, show an image of the visual canvas with the selection that was used, saves the current selection to the cloud, shows the list of genes in the selection (along with the count of genes, figure 4.3), performs a differential expression analysis on the selected selections (figure 4.4), saves the current selection to the local computer, edits the name and details of the current selection, and remove the current selection. Selections are saved as tab separated text files. The show snapshot option shows a screenshot of the image when the selection was made. This is useful if the image was zoomed or panned to show a specific area.

When running in online mode, selectiond can be saved to the cloud and previously saved selections are shown and can be re-selected.

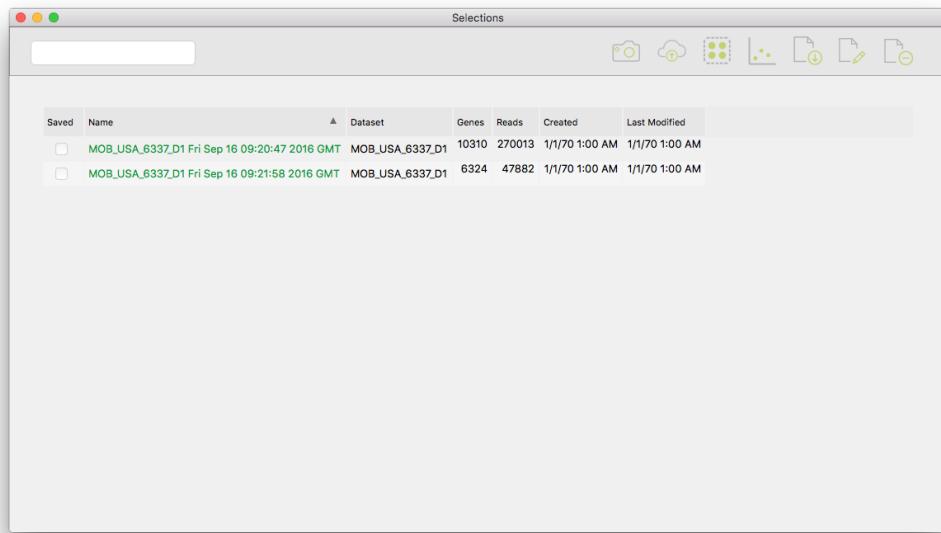


Figure 4.2: The selection window.

Gene	Total count
ENSMUSG00000000001	14
ENSMUSG00000000056	3
ENSMUSG00000000058	7
ENSMUSG00000000085	33
ENSMUSG00000000088	183
ENSMUSG00000000093	1
ENSMUSG00000000094	1
ENSMUSG000000000126	1
ENSMUSG000000000131	8
ENSMUSG000000000134	9

Figure 4.3: The genes in the current selection.

4.2 Differential Expression Window

The differential expression window shows a log-log graph of counts for each gene in both selections, a list of all the genes and their counts in both selections and some statistics for the differential expression analysis. The statistics include The number of genes in both selections, the number of genes present in both selections, and the number of genes unique to each selection. Also displayed is the Pearson correlation for the differential expression. It is also possible to change the threshold for the number of reads used in the differential expression analysis. Changing the threshold changes both what is displayed on the graph, and also recomputes the Pearson correlation. The save button will save the current graph to the local computer.

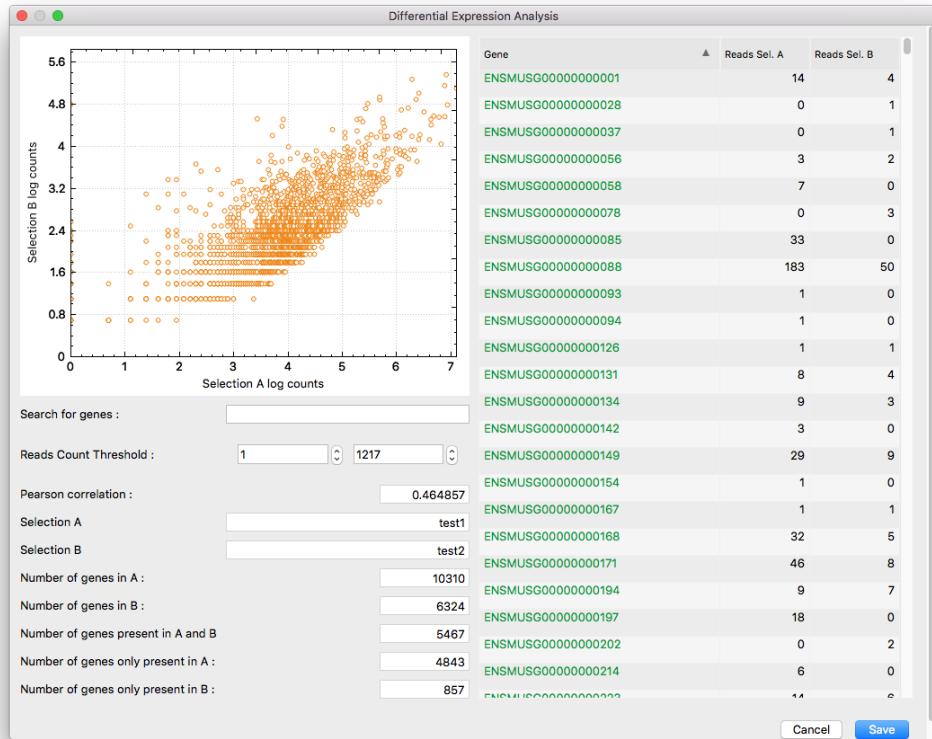


Figure 4.4: The differential expression window.

Chapter 5

Known Issues and Future Plans

5.1 Known Issues

As the application is still in the early alpha phase it may contain functional as well as visual discrepancies and issues. Some of these are known but were out of the scope of this release.

Below follows a list detailing each known issue, the afflicted platform as well as a brief description of its effect:

- General
 - Speed and memory issues
- Linux
 - Some GUI related artifacts
- Windows
 - Some GUI related artifacts

Appendices

Appendix A - Installing the viewer

Installing in OSX

- Download and install Qt from: <http://qt-project.org/downloads>
- Download and extract QCustomplot sources from <http://qcustomplot.com/>
- Install the necessary dependencies by running the following code (assuming MacPorts is installed)
`sudo port install xcode cmake git`
NOTE: Make sure the XCode command line tools are installed.
- Clone the repository and build the application. Remember to tell CMake where QCustomplot is using DCMAKE_PREFIX_PATH.

`git clone https://github.com/SpatialTranscriptomicsResearch/st_viewer /path/to/source
mkdir /path/to/build
cd /path/to/build
cmake [-DCMAKE_INSTALL_PREFIX="/usr/local/bin"] \
[-DCMAKE_BUILD_TYPE="Debug | Release"] \
[-DCMAKE_PREFIX_PATH="/path/to/libraries"] \
[-DCMAKE_OSX_SYSROOT="/path/to/macosx.sdk"] \
[-DCMAKE_OSX_DEPLOYMENT_TARGET=version] \
[-DPUBLICKEY="path_to_ssl_key"] \
[-DCONFIG_FILE="path/to/network_config_file"]
/path/to/source`

Where:

DCMAKE_INSTALL_PREFIX is where to install the viewer.

DCMAKE_BUILD_TYPE indicates the type of build (Debug or Release, default is Release).

DCMAKE_PREFIX_PATH is the location of any extra files (e.g. binaries for Qt5 and the source for QCustomplot).

DCMAKE_OSX_SYSROOT is the path to the MacOSX SDK that is to be used (e.g. /Applications/Xcode.app/Content).

DCMAKE_OSX_DEPLOYMENT_TARGET is the target OSX version (can be older than the version of OSX running on the computer).

DPUBLIC_KEY [optional] is the public key to access datasets in the online database.

DCONFIG_FILE [optional] is the config file that is needed to access datasets in the online database. This file contains the endpoints and OAuth access settings. This file is loaded at runtime and is able to be edited after the viewer is installed.

- Build the application.

```
make -j8
```

- Alternatively, you can build a standalone DMG bundle that you can install and/or distribute.

```
make dmg
```

Installing in UNIX

- Download and install Qt from: <http://qt-project.org/downloads>
- Download and extract QCustomplot sources from <http://qcustomplot.com/>
- Open a terminal and issue the following commands (apt-get in Ubuntu, yum in Fedora etc.)


```
sudo apt-get install cmake git ubuntu-dev-tools
sudo apt-get install libglu1-mesa-dev freeglut3-dev mesa-common-dev
```
- Clone the repository and build the application. Remember to tell CMake where QCustomplot is using DCMAKE_PREFIX_PATH.

```
git clone https://github.com/SpatialTranscriptomicsResearch/st_viewer /path/to/source
mkdir /path/to/build
cd /path/to/build
cmake [-DCMAKE_INSTALL_PREFIX="/usr/local/bin"] \
[-DCMAKE_BUILD_TYPE="Debug | Release"] \
[-DCMAKE_PREFIX_PATH="/path/to/libraries"] \
[-DCMAKE_OSX_SYSROOT="/path/to/macosx.sdk"] \
[-DCMAKE_OSX_DEPLOYMENT_TARGET=version] \
[-DPUBLICKEY="path_to_ssl_key"] \
[-DCONFIG_FILE="path/to/network_config_file"]
/path/to/source
```

Where:

DCMAKE_INSTALL_PREFIX is where to install the viewer.

DCMAKE_BUILD_TYPE indicates the type of build (Debug or Release, default is Release).

DCMAKE_PREFIX_PATH is the location of any extra files (e.g. binaries for Qt5 and the source for QCustomplot).

DCMAKE_OSX_SYSROOT is the path to the MacOSX SDK that is to be used (e.g. /Applications/Xcode.app/Content).

DCMAKE_OSX_DEPLOYMENT_TARGET is the target OSX version (can be older than the version of OSX running on the computer).

DPUBLIC_KEY [optional] is the public key to access datasets in the online database.

DCONFIG_FILE [optional] is the config file that is needed to access datasets in the online database. This file contains the endpoints and OAuth access settings. This file is loaded at runtime and is able to be edited after the viewer is installed.

- Issue the following commands to build and install the viewer.

```
make -j8
make install
```

- To build a standalone package to install and/or distribute:

```
make package
```

Installing in Windows

There are many different ways to build the ST Viewer in Windows. Here Cygwin is used.