

Spotxel® 1.7

Microarray Image and Data Analysis Software

User's Guide

27 April 2017 - Rev 7

Spotxel® is only intended for research and not intended or approved for diagnosis of disease in humans or animals.

Copyright 2012-2017 Sicasys Software GmbH. All Rights Reserved.

Sicasys Software GmbH
Hans-Bunte-Str. 8
69123 Heidelberg
Germany

Phone +49 (62 21) 7 28 50 40
Fax +49 (62 21) 7 28 48 94
Email info@sicasys.de
Web www.sicasys.de

Table of contents

1	Introduction.....	1
1.1	Installation.....	1
1.2	Product Activation.....	1
1.3	Software User Interface	2
1.4	Terms and Concepts.....	3
1.5	Microarray Data Analysis.....	4
2	Preparation for Microarray Data Analysis.....	5
2.1	Loading Data.....	5
2.2	Array Alignment.....	6
2.3	Spotxel Project File	6
3	Image and Array Processing	7
3.1	Image Processing.....	7
3.2	Image and Array Rotation	8
3.3	Array Alignment Options.....	9
4	Quantification of Microarray Data	11
4.1	Quantifying Microarray Data.....	11
4.2	Change of Intensity Values.....	12
4.3	Spot Detection Methods	13
4.4	Noise Processing.....	14
4.5	Spot Detection Options	17
4.6	Background Correction.....	17
5	Scatter Plot & K-Means Clustering	19
6	Batch Processing.....	21
7	Data Preprocessing and Normalization.....	22
7.1	Datasets.....	23
7.2	Replicate Processing.....	24
7.3	Data Filtering	25
7.4	Data Normalization	26
8	Data Mining Tools.....	27
8.1	Principal Component Analysis (PCA)	27
8.2	Hierarchical Clustering Analysis (HCA)	29
9	Product Activation.....	30
10	End-User License Agreement	32
11	Index.....	34

1 Introduction

Spotxel® software supports microarray image and data analysis. This includes microarray image analysis and automatic batch processing of a number of microarray images. You can then improve the quality of microarray data with replication processing, data filtering, and data normalization tools. Furthermore, you can discover features and samples that influence the study and their relationship with data mining tools.

1.1 Installation

Spotxel® runs natively on Windows and Mac OS X platforms. Installation of the software requires rights of a system administrator.

Hardware Requirement

Minimum hardware: 1.5 GHz processor, 2GB RAM.

Recommended hardware: 2.0 GHz Dual-Core or more powerful processor, 4 GB or more RAM.

Windows Platforms

Spotxel® works on Windows XP, Windows 7, Windows 8, and Windows 10. We recommend the 64-bit version of Spotxel for 64-bit Windows computers. Check your Windows version as described [here](#).

Simply run the Spotxel® setup to install the software. If the current Windows account is not an administrator, you will be asked to input an administrative account and its password.

Mac OS X platforms

The software runs on Mac OS X 10.7 and later versions. Unzip the package and double-click on the .pkg file to launch the installer. During the installation you will be prompted to provide a system administrator's account and password. Upon completion, Spotxel® is installed in the */Applications/Spotxel* folder.

1.2 Product Activation

After installing Spotxel® on Windows, you need to activate the software with a *trial serial number* obtained from the software provider or its distributor. This enables the use of Spotxel® with full functionality for 14 days. The trial use for Spotxel® on Mac OS X platforms is handled automatically and does not require this step.

When the free trial time has expired, you can [buy](#) a software license to continue using Spotxel®. Upon the purchase, you receive a serial number and use it to activate the license. Please refer to the product activation steps in Section 9 or at the online help [here](#).

1.3 Software User Interface

Related software controls are grouped in labeled components as shown in Figure 1. We refer to a software component using the name listed in Table 1.



Figure 1: The Software User Interface.

Component	Component Name
1	The menu
2	The canvas toolbar
3	The main toolbar
4	The control panel
5	The canvas
6	The <i>Spot Image</i> widget
7	The table of quantified data

Table 1: Software Components.

The main toolbar enables quick access to a group of related functions. They are described in Table 2. Clicking on a button on the main toolbar opens the control panel for the function group. The

software shows the data and the analysis results in the corresponding sheet on the right of the control panel. E.g., Figure 1 shows the control panel with controls and settings for data quantification. On the right side, the *Imaging* sheet displays a microarray image in the *canvas*, the signal of the currently selected spot in the *Spot Image* widget, and the *table of quantified data*.


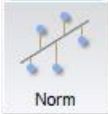

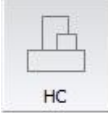

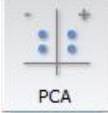


 Images	Select image channel. Change image's intensity. Rotate images.	 Norm	Process replicates, filtering data, and normalize data.
 Arrays	View properties of blocks and spots. Rotate and move blocks.	 HC	Hierarchical Clustering Analysis: Show features and samples on a heat map with their correlation.
 Quantify	Quantify the array data. Browse the quantified data and export it to CSV and GPR files.	 PCA	Principal Component Analysis: Select important features and samples.
 Scatter Plot	Show spots in a scatter plot according to their quantified data.	 Batch	Batch Processing: Automatically process and quantify many microarray images.

Table 2: The Main Toolbar and Related Functions.

1.4 Terms and Concepts

In this manual the term *array* is used to refer to the spot layout and annotation of a microarray. We assume that the array is saved as a *GenePix Array List (GAL) file*. The term *image* or *microarray image* is used to denote a scanned image of the printed microarray.

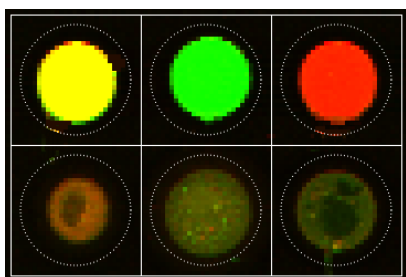


Figure 2: A rectangular block with 6 spots.

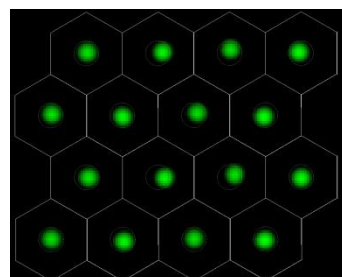


Figure 3: A hexagonal grid with 16 spots.

An array consists of blocks. Each *block* is a group of spots located next to each other. In the canvas, a *spot* is illustrated as a *white square* in a *rectangular block*, or a *white hexagon* in a *hexagonal block*. Figure 2 shows a rectangular block with two rows and three columns. A hexagonal block is presented in Figure 3; here sixteen spots are arranged in four rows and four columns. Within each spot, the *spotted region* is defined as the area bounded by the *dashed circle*. Its diameter is specified by the *spot diameter* parameter for each block.

The binding signals of a microarray tested with a sample are converted by a microarray scanner into a digital *array image* containing a matrix of *pixels*. Each pixel has a gray value representing *pixel intensity*. Array images are often saved in the *TIFF format*¹. In *8-bit grayscale images*, the gray value ranges from 0 to 255. This value can be from 0 up to 65535 in *16-bit grayscale images*. Since they have a broader range of signal levels than the 8-bit format, 16-bit grayscale images are recommended for array image analysis.

Quantification is the procedure that estimates the true binding signal for each spot and represents its signal value in terms of statistic measurement of pixel intensities within that spot. Obviously, the quantification quality depends on the *spot finding* or *spot detection* procedure which determines which pixels in the array image should belong to a spot in the array. *Background correction* also contributes to the quantification quality. It estimates signals caused by non-specific binding and removes them from the spot's signal.

For each spot in the array, the median and the mean of its *raw*, *background*, and *foreground* values are calculated. *Raw* represents the intensity value of the spot's signal. *Background* is the estimated value of the signal caused by non-specific binding. The value of interest is *foreground*; it is computed by subtracting the background value from the raw value.

Array alignment is the process of associating spots in the array with their signal in the image. The spot's signal is presumably due to the binding of immobilized substance in the spotted region with the sample. Therefore, before quantification we will reallocate the array such that the spotted regions are as close to the spots' signal as possible.

1.5 Microarray Data Analysis

From the software perspective, typical tasks of microarray data analysis include:

1. Quantification of microarray data
 - Load the scanned images and the array file.
 - Align the array to the images.
 - Quantify the microarray data.
2. If necessary, batch processing of multiple microarray images, i.e. the automation of step 1.
3. Preprocess the data with the data filtering and normalization tool.
4. Discovery of parameters influencing the study and their relationship with data mining tools.

The following sections explain how to accomplish these tasks with Spotxel®.

¹ <http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf>

2 Preparation for Microarray Data Analysis

2.1 Loading Data

To analyze the microarray data, two input data are required:

- Scanned images of the microarray in the TIFF format.
- The array file prepared in the GAL format (*.gal).

Spotxel® supports 8-bit or 16-bit grayscale images or 24-bit color images. For the best image quality 16-bit grayscale TIFFs are recommended.

Loading Input Data

- Click *the Images > Open Image* menu and select the microarray image file. For grayscale images, select to display each image with either the *Red* channel or the *Green* channel.
- Click the *Arrays > Open Array* menu and select the GAL file.

After being loaded, the images and the array are shown in the graphical canvas (Figure 1). To obtain an appropriate view, you can use the *Zoom In* and *Zoom Out* buttons on the canvas toolbar or select a predefined zoom level in the *Zoom* combo-box. Alternatively, an arbitrary value can be entered directly into the *Zoom* combo-box.

Block

Name

Number of columns

Number of rows

Spot spacing

Horizontal (µm)

Vertical (µm)

Spot diameter (µm)

Top-left spot's position

Horizontal (µm)

Vertical (µm)

Spot

Spot

Block

ID

Name

Spot position

Local Column

Local Row

Horizontal (µm)

Vertical (µm)

(a) Block Properties

(b) Spot Properties

Figure 4: Properties of an Array Object.

Viewing Array Data

In the *Array Object* section of the *Arrays* control panel, you can view properties of a block or a spot. To view a block's properties (Figure 4-a) in the *Block* page, first open the *Block* page by clicking on it,

and then hover the mouse over the block in the canvas. Similarly, you can open the *Spot* page and then points to a spot to view its properties (Figure 4-b).

2.2 Array Alignment

As mentioned in Section 1.4, the array needs to be aligned with the image before quantifying the microarray data. This can be done automatically. You can also manually align the array with the image, probably after processing the image or the array, e.g. rotating a block or increasing the spot's visibility. Please refer to Section 3 for image and array processing functions.

Aligning Array Automatically

- Click the *Align Array* button in the *Arrays* control panel.

There are many options that can be adjusted to optimize the alignment of your arrays. They are detailed in Section 3.3.

Aligning Array Manually

- Click *Ctrl-A* to select all blocks in the array. To select individual blocks, click on them while pressing the *Ctrl* key.
- Click on the selection and drag the corresponding blocks to the intended position.

The aligned position of the blocks in the array can be saved with the *Arrays > Save Array* menu. In addition, you can save the array to another GAL file using the *Array > Save Array As* menu.

2.3 Spotxel Project File

It is recommended that the analysis of each microarray image be saved to a Spotxel® project file (*.spotxelproj) using the *Project > Save Project* menu. The saved data includes the path to the image, the aligned array, and the quantified data. When opening the project file with the *Project > Open Project* menu, the software will load all the saved data. This enables to manage all the analysis data for one microarray image with a single project file. In addition, later you can use these project files directly with data mining tools.

The paths to the microarray image, the GAL file, and the project file are shown in the *Data Files* section of the *Quantification* control panel.

3 Image and Array Processing

3.1 Image Processing

Adjusting contrast and brightness

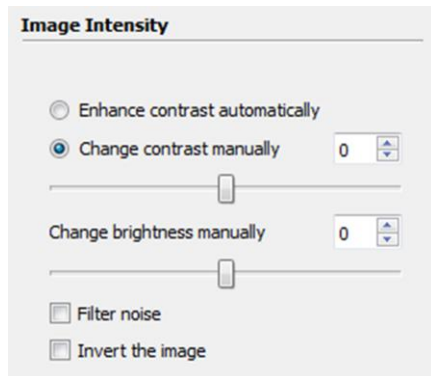


Figure 5: Image Processing.

This process can make the spots more visible and ease the array alignment. These functions are available in the *Image Intensity* section of the *Images* control panel.

- Choose the *Enhance contrast automatically* option to maximize the spot visibility.
- You can manually adjust the image's brightness and contrast by moving the slider. Alternatively, a value between -99 and 99 can be entered directly.

Colorization

You can also increase the spot visibility by means of the *colorization* tool. This depicts a grayscale image, or a single channel of a color image, as a color image in which a pixel's color is determined by its original intensity value. Figure 7 illustrates the effect of the tool. It can be seen that many spots of weak signal have become visible in the processed image.

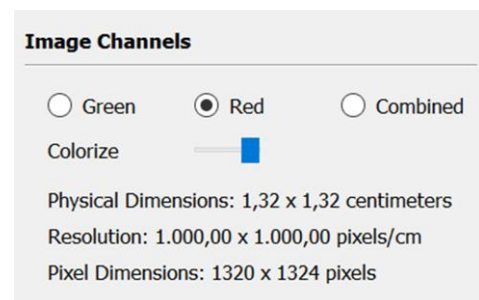
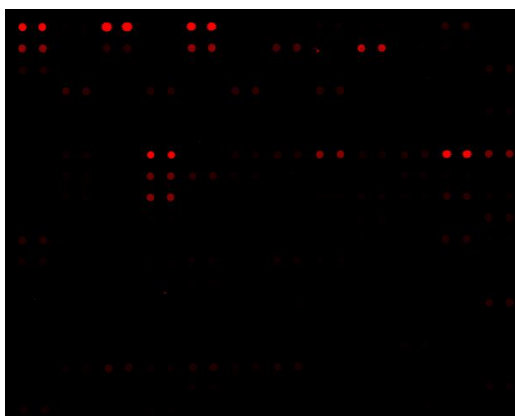
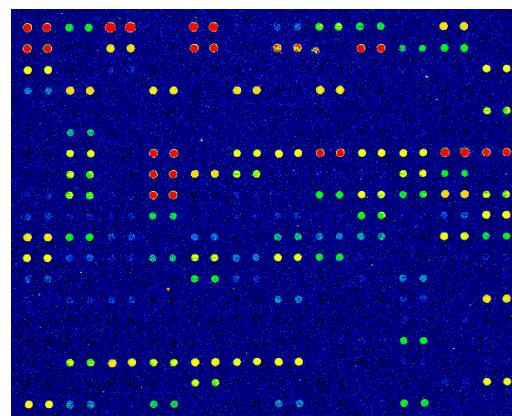


Figure 6: Colorization options.



a) Original image.



b) Image after colorization.

Figure 7: Effect of Colorization.

The tool can be turned on and off by clicking on the left or right side of the *Colorize* switch (Figure 6). Note that a single channel, either *Green* or *Red*, must be selected first to enable the switch.

For each pixel in the original image, the tool first determines the range that the pixel intensity belongs to, and shows the pixel in the corresponding color. The number of colors is fixed, while the ranges of intensity values are predefined (Figure 8) and adjustable. To change the setting, click the *Configuration > Colorization Setting* menu.

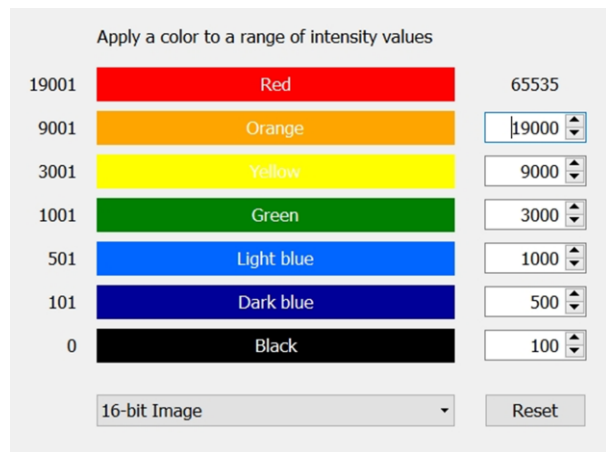


Figure 8: Colorization Setting.

Noise Filtering

Noise in the loaded image(s) can be reduced by selecting the *Noise Filtering* check-box.

Inversing

Select the *Inversion* check-box to create the negative image. This is useful when processing the image of colorimetric assays in which the stronger the signal, the darker the pixel.

3.2 Image and Array Rotation

Rotating images

You can flip and/or rotate images at angles of 90°, 180°, or 270°. These functions are located in the *Image Rotation* section of the *Images* control panel (Figure 9-a).

Rotating Array

If the array slightly deviates from the image at a small angle, it is recommended that the array (i.e. related blocks) be rotated, since image rotation may change the image data. You can select blocks and rotate them at an arbitrary angle. These functions are located in the *Array Rotation* section of the *Arrays* control panel (Figure 9-b). The degree change can be as small as 0.01°.

Selected blocks can be rotated in clockwise or counterclockwise direction, with a rotation center defined as follows:

- *Global*: the top-left of the image.
- *Local*: the top-left of each block.

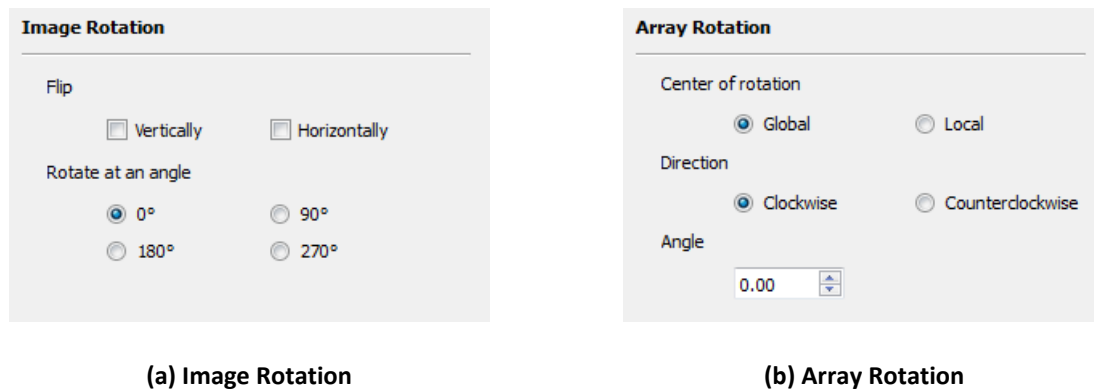


Figure 9: Rotation of Images and Arrays.

3.3 Array Alignment Options

The automatic function may not align the grids with the image precisely in the first place. One reason is that the slide image is somewhat rotated when scanning. Consider the first column of an array, a small rotation of 0.25° of the slide image would result in different horizontal position of spots in the first rows and those in the last rows. It could also be due to the spotting process that the blocks' position inconsistently deviates from their expected horizontal and vertical coordinate. You can fine-tune the alignment function to handle these cases by clicking the *Configurations > Array Alignment Options* menu. The setting dialog, whose first part is shown in Figure 10 and the second part in Figure 11, is then activated.

Alignment Signal

This setting (Figure 10, the *Alignment Signal* section) is useful when analyzing dual-color images. It enables to specifically set from which channel (Red and/or Green) the signal is used for array alignment. The default option is *Combined*; the software takes the strongest signal among two channels for aligning the array.

For single-color analysis, naturally the alignment uses the only signal from the loaded channel.

Alignment Controls

In addition to selecting signal from a channel for aligning array, you can specifically use some positional controls to optimize the alignment. This consists of following steps:

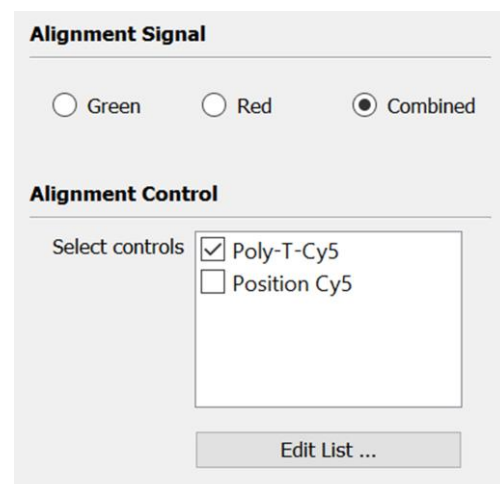


Figure 10: Alignment Options - Signal & Controls.

- To employ one or more controls for the alignment, check the box next to them. In the example in Figure 10, the control *Poly-T-Cy5* is selected as an alignment control.
- To add a new control or modify an existing one, click on the *Edit List* button in the *Alignment Control* section. The software then shows a list of all controls.
- To create a control, click the *Add* button. To modify an existing one, click the *Change* button. In the *Edit the Control* dialog, specify the name, the **ID**, and the replication mode of that control spot. Once created, the controls remain for next uses of the software.

In the array, spots that have Name (in the GAL file) being the ID of the selected alignment controls will be used as the positional markers. The array alignment will then position the grids in such a way that best match these positional markers, instead of all spots in the array.

Detect Rotation

Check this option (Figure 11, the Orientation section) to detect if the slide image is rotated and process the alignment accordingly. Please note that the alignment will takes a bit longer if rotation detection is included.

Currently, the alignment function limits the rotation angle between -0.50° and 0.50° (i.e., between 0.50° clockwise and counter-clockwise).

Orientation

☒ Detect rotation

Horizontal distance between

Left-most blocks 0,50

Two adjacent blocks 0,75

Vertical distance between

Top-most blocks 0,50

Two adjacent blocks 0,75

Figure 11: Alignment Options – Orientation.

Offset of Blocks' Position

Normally, two adjacent blocks are planned to have approximately the same horizontal (X) or vertical (Y) position. The printed blocks, however, may slightly offset from that planned position. In Figure 12, we observe the difference in the horizontal position of block 1 and block 3 as well as in the vertical position of block 3 and block 4.

To control the offset of blocks that should have the same X- or Y-position, you can set the following parameters (Figure 11, the *Orientation* section):

1) *Horizontal distance between* (measured as ratio to the spot width)

- *Left-most blocks*: The maximum difference in X-position between left-most blocks of the array. In Figure 12, these are block 1 and block 3.
- *Two adjacent blocks*: The maximum difference in X-position between two adjacent blocks which are not left-most. In Figure 12, these are block 2 and block 4.

2) *Vertical distance between* (measured as ratio to the spot height)

- *Top-most blocks*: The maximum difference in Y-position between top-most blocks of the array. In Figure 12, these are block 1 and block 2.
- *Two adjacent blocks*: The maximum difference in Y-position between two adjacent blocks which are not top-most. In Figure 12, these are block 3 and 4.

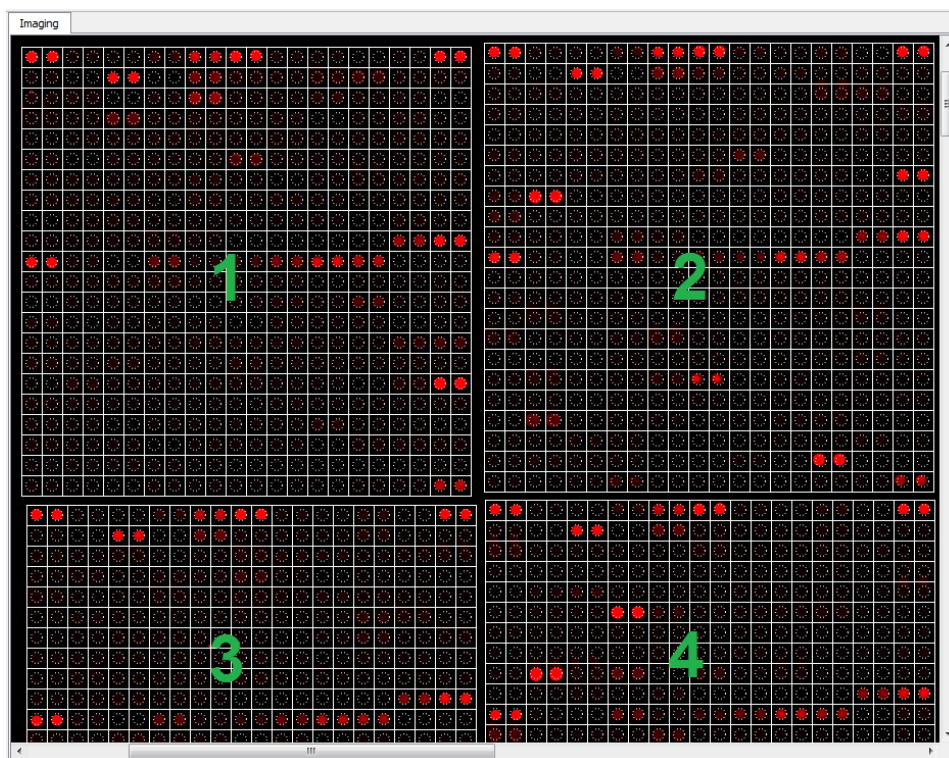


Figure 12: Offset of Blocks.

4 Quantification of Microarray Data

4.1 Quantifying Microarray Data

Click the *Quantify* button in the main toolbar to activate the *Quantification* control panel.

- Click the *Quantify Array* button to get the data quantified for the entire array.
- To quantify some blocks, select them and then click on the *Quantify Selection* button.

View Image and Array Data during Quantification

The quantification procedure may take long, depending on the image resolution and the array size. Fortunately, the software supports multi-threading, so that while waiting for the quantification to be finished, you can still navigate through the image and the array data. This includes scaling and scrolling the image, as well as viewing properties of blocks and spots, as described in Section 2.1.

Quantified Array Data

For each spot in the array, the median and the mean of its *raw*, *background*, and *foreground* values in each channel (Red and/or Green) are calculated. If the raw value of a spot is smaller than its background value, the spot is flagged *Error* and its foreground value is set to zero. (Section 4.3 details the methods used to calculate the raw value and the background value.)

In the *Imaging* sheet, you can view the spots and their quantified data simultaneously (Figure 13). The image part corresponding to the selected spot and its neighbors are displayed in the *Spot Image* widget. In addition, selecting a row in the table of quantified data highlights the corresponding spot in the canvas. This also opens the *Spot* page in the *Array Object* section in the *Arrays* control panel and shows the spot's properties there. Similarly, when the *Spot* page is opened, clicking on a spot in the canvas will highlight its quantified data in the table.

You can also browse the spots' quantified data in the *Quantified Data* sheet, which shows only the table of quantified data and the *Spot Image* widget. In both *Imaging* and *Quantified Data* sheets, you can export the quantified data to a CSV file (*.csv) or a GenePix Result file (*.gpr) for further analysis.

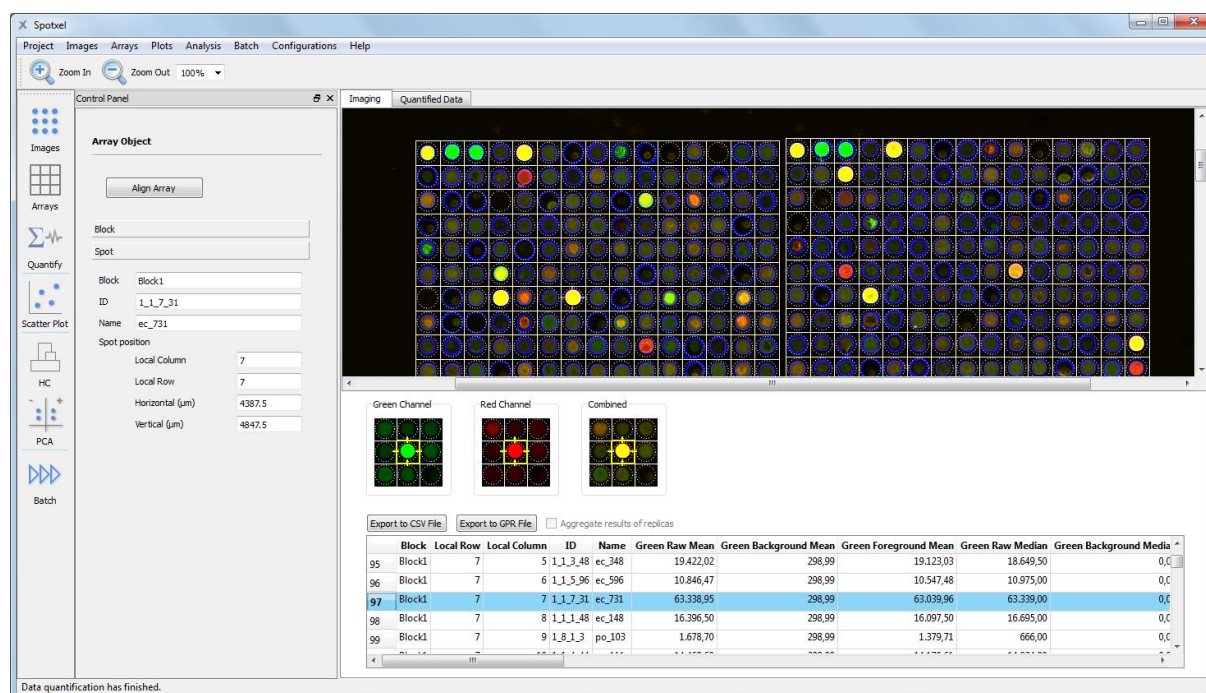


Figure 13: Quantified Data.

4.2 Change of Intensity Values

During manual array alignment, you may have adjusted the contrast and/or the brightness of the images to make the spots visible. This changes the image data and alters the analysis results. It is

recommended that *the original image data be used for the quantification*. Therefore, by default *the quantification procedure uses the original image data*, i.e. it *excludes* changes made to the image by inversion, noise filtering, and adjustment of contrast and/or brightness.

This option can be intentionally changed as follows. In the *Quantification Options* section of the *Quantification* control panel, select *Yes* for the *Include change of the images' intensity value* option (Figure 14). To use the original image data for the data quantitation again, select *No*. Please note that when the *No* option is chosen, the *Spot Image* widgets in the *Imaging*, *Quantified Data*, and *Scatter Plot* sheets show the spot's counterpart image using the original image data, even though the image's contrast or brightness has been changed.

4.3 Spot Detection Methods

In the *Quantification* control panel, the *Quantification Options* section (Figure 14), the *Spot detection method* option defines how a spot's raw value and background value are calculated.

Figure 14: Quantification Options.

- *Fixed-Spot*: The software *always* uses the pixels in the *spotted region* (Figure 15, dashed circles) to compute the raw value. The background value is calculated based on the pixels in the remaining region within the spot.

- *Flex-Spot*: This method can flexibly detect the spot's signals even though their shape, size, and position are not in accordance with the spotted region. The *detected spot border* is shown in blue. Calculation of the raw value is based on pixels *within* the blue border. Pixels in the remaining region within the spot are used to calculate the background value.

The Flex-Spot method is recommended because it does not require the spots in the image to rigidly match with their spotted regions, as specified in the GAL file. Figure 15 illustrates such a case. Here, within a spot (the white square) the spotted region is depicted by the white dashed circle. The spots' signals, shown in red, have different shape and position from those of their spotted region. Despite that fact, the Flex-Spot method can still precisely find the spots and highlight their border in blue.

For *hexagonal* (or *orange-packing*) blocks, the same principle is applied. The only difference is that the spot is a white hexagon in this case, instead of a white square as explained above.

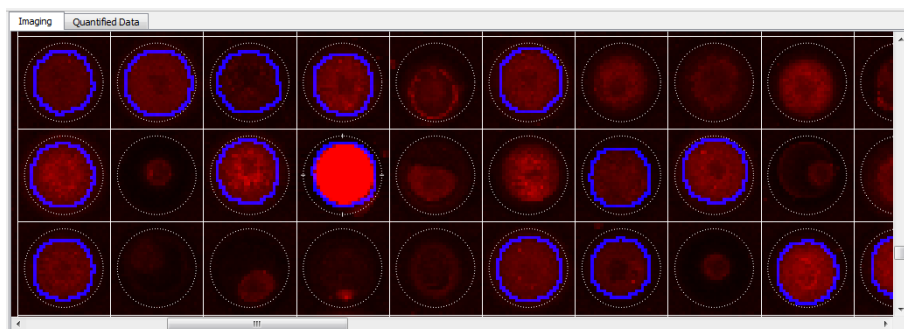


Figure 15: The Flex-Spot Method.

Please note that the above described mechanism for background calculation uses only the background pixels within the spot. It is thus called the *Local method* or *local background correction*. You can also have background values based on pixels from any region of the image by means of background controls.

4.4 Noise Processing

Employ Noise Processing during Quantification

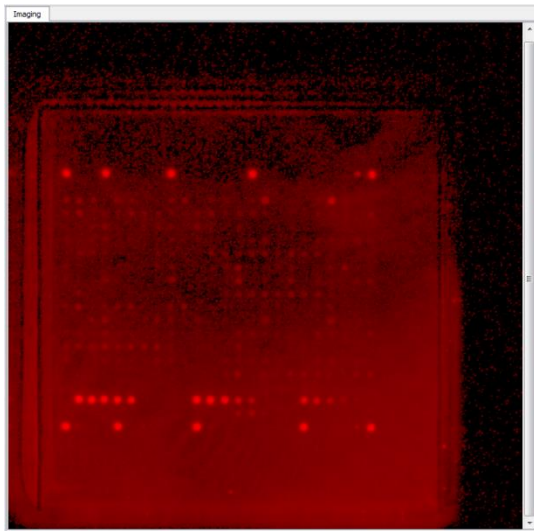
By checking the *Process noise* option (Figure 14), you can include noise processing during the quantification procedure. This includes the processing of background noise, foreground noise, and signal due to smearing.

Since noise processing effectively reduces background noise, it can be regarded as an (implicit) background correction method. Please note that you can *process noise* during quantification with both *Fixed-Spot* and *Flex-Spot* methods.

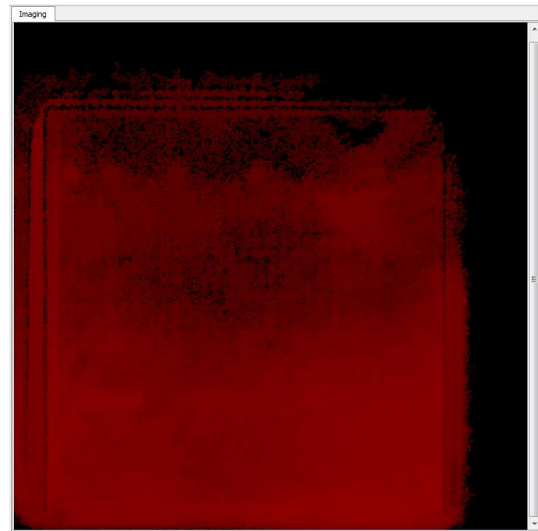
Background Noise

Noises in microarray images can mislead the spot detection procedure and result in wrong estimation of true specific-binding signal. Figure 16-a show a slide image in which the background signal is not homogenous across the slide. Furthermore, it conceals lots of specific-binding signal. This badly affects the array alignment as well as the spot finding.

Equipped with powerful image processing functions, the software is able to process such background noise. Figure 16-b visualizes the detected background signal alone, while Figure 16-c shows the slide image again but having the background signal eliminated. We can now see the spots' signal clearly, even though many of them are largely hidden by the background noise in the original image.



(a) Original image.



(b) Detected background signal.

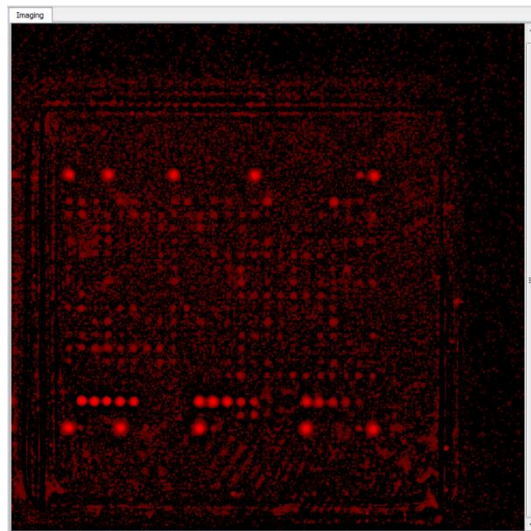
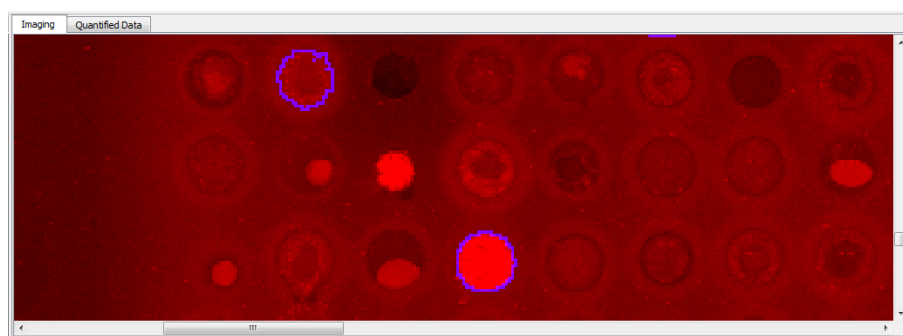
(c) The *Raw* signal: The image with background signal eliminated.**Figure 16: Noise Processing.****Figure 17: Background Noise Processing.**

Figure 17 shows another slide image with background noise spanning across the whole slide. We want to “remove” the background layer so that only the meaningful spot signal remains. Furthermore, in

this case signal with small shape is due to non-specific binding and should not be selected either. As can be seen, only two desired spots are selected, having their border highlighted in blue.

Foreground Noise

Figure 18 illustrates a slide image with “foreground” noise. Here, the two large red bands are clearly due to non-specific binding and their signal should not be part of any spot. Based on the spots’ shape and size, the software can process such foreground noise; it selects only valid spots (whose signal also in red) and highlights them with a blue border then.

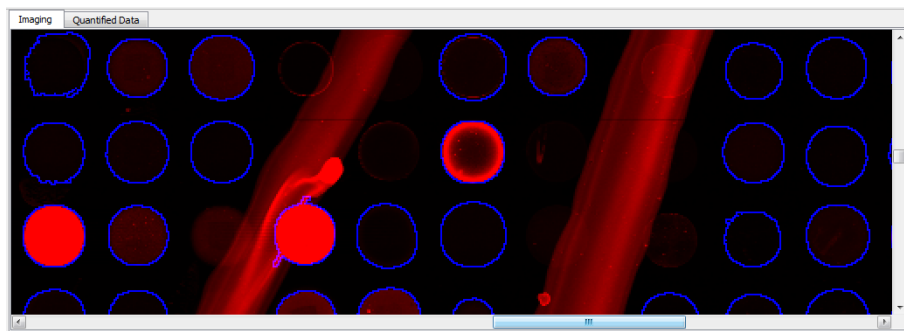


Figure 18: Foreground Noise Processing.

Smearing Signal

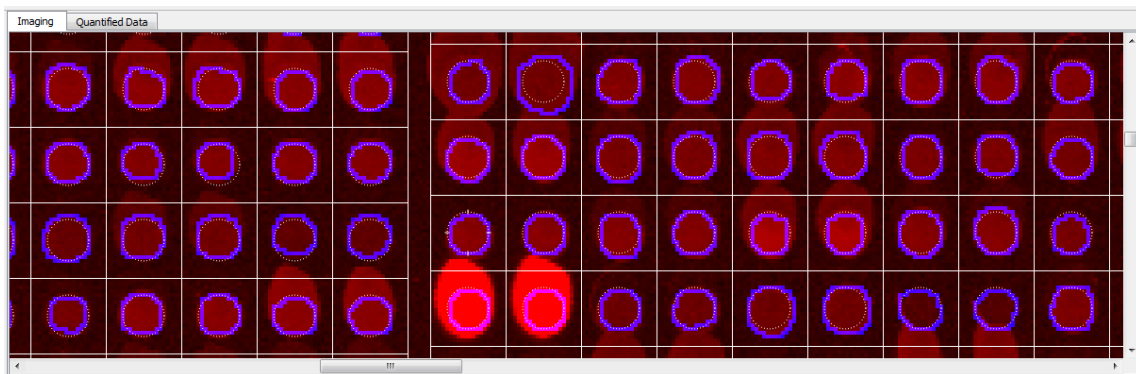


Figure 19: Smearing Processing.

Figure 19 presents a case in which the non-specific binding signal is due to smearing. Note that the contrast of this slide image is increased for ease of illustration, and the spot signal is in red. Based on the expected shape and position of valid spots, the software is able to detect the case and select only signal due to the local binding within each spot. The border of detected spots is also highlighted in blue, showing that the spot signal is not affected by smearing from neighboring spots.

4.5 Spot Detection Options

Smallest Spot Size

You can set the size limit of a “valid” spot by means of the *Smallest spot size (%)* parameter (Figure 14). Suppose that this value is 50%. Imagine a virtual square whose side length being 50% or half of the spot diameter. If the spot signal is smaller than or can be contained in that virtual square, the Flex-Spot method will reject this spot. The software will then use the Fixed-Spot method to compute the raw value for that spot.

As an example, the spots shown in Figure 15 are detected by the Flex-Spot method with the *Smallest spot size (%)* parameter being 50%. It can be observed that small spots do not have a blue border.

Show Border

After data quantification, you can turn the detected spot border on and off using the *Show border* option (Figure 14). The software supports saving the border information in the project file (*.spotxelpoj). Therefore, you can still observe the detected spot border when reopening the project file. This is particularly useful for reviewing the quantified data generated by batch processing.

Undetectable by Flex-Spot

If the Flex-Spot method cannot detect a spot due to e.g. being smaller than the size limit, weak signal, or noisy data, the software employs the Fixed-Spot method to compute the raw value. *No blue border* is shown within the spot if the Fixed-Spot method is used.

4.6 Background Correction

Background correction methods can be selected in the *Quantification Options* section of the *Quantification* control panel. By default, the *Local method* is employed.

Local Method

Spotxel® supports different levels of local background correction (Figure 20). The default one for GAL files is the *block level*.

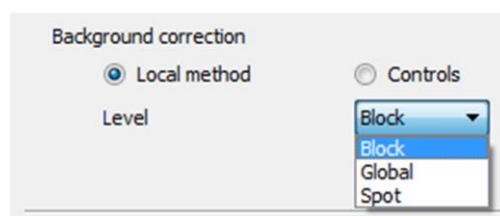


Figure 20: Background Correction Options.

At the *block level*, all spots in a block will have the same background value. It is computed as follows. First, a list of background pixels of all spots in the block is created. The mean and the median of the pixel intensity values in this list are then used as the background value.

You can choose to have a global background value for the *entire array* by selecting the *global level*. The calculation is based on the background pixels of all spots in the array. At the *spot level*, a spot's background value is based on its background pixels only. Therefore, it is likely that the background values are different between spots.

Background Controls

Instead of having the background values locally calculated, you can use a background control to explicitly specify the image region from which the background value is calculated. You can define a *global background value for the entire array* using a background control as follows.

- Open the *Block* page in the *Array Object* section of the *Arrays* control panel.
- Right-click at a point in the image where the pixels represent the background for the spots. Choose *Create Background Control*. Specify the shape and the number of spots.
- Perform data quantification. All spots in the array now have the same background value which is obtained from the background control. We call that an “*association*” between the blocks and the background control.

In addition to having a global background value for the entire array, you can flexibly choose *an individual background value for each block*. This is done by creating a background control just for the block and establishing the association between them.

- Right-click on the block to popup the context menu. Choose *Background Control > Remove Association* to release the block from the first background control.
- Right-click on the block again and choose *Background Control > Establish Association*. Then click on the background control with which you want to associate this block. You can check this association using the *Highlight Associated Control* context menu (Figure 21).

Please note that the background correction method is automatically set to *Controls* after the creation of the first background control. In the *Quantification Options* section of the *Quantification* control panel, you can select *Local method* again for the *Background correction* option (Figure 20).

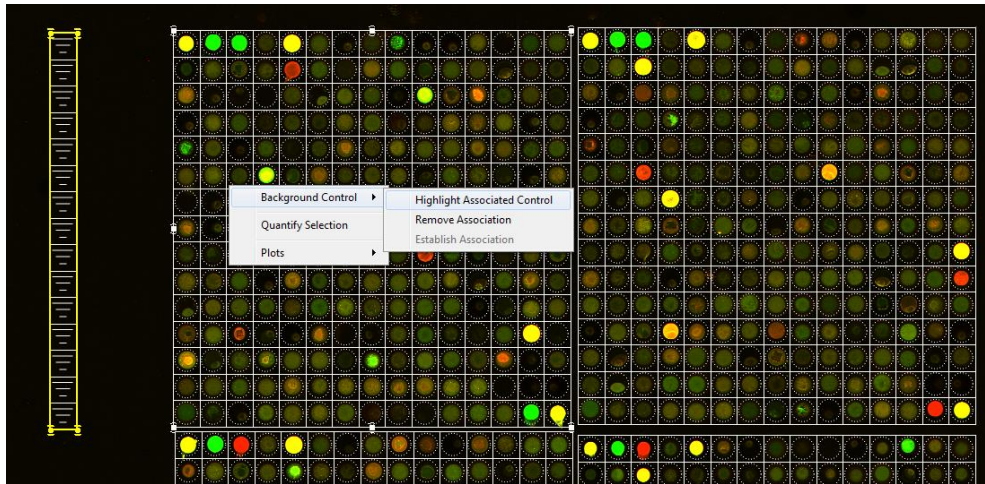


Figure 21: Associated Background Control.

5 Scatter Plot & K-Means Clustering

Scatter Plot

After quantifying the data, you can depict the microarray data on a two-dimensional scatter plot (Figure 22). This enables to visually examine and select spots according to their quantified values.

To start with, click on the *Scatter Plot* button in the main toolbar. Initially, spots of all blocks in the array are shown. You can limit the plot to a certain block using the *Blocks* list-box. The *X*- and *Y*-axes can be any quantified value. By means of the two blue *threshold bars* on the plot, you can select spots whose *X*-values are between the two threshold values. These spots are then populated into the table below the plot. By clicking on or hovering over a spot in the plot, you can view the signals of the spot and its neighbors in the *Spot Image* widget as well as its properties in the *Spot Details* widget. This also highlights the spot's quantified data in the table.

You can export the table's data to a CSV file. It can be either the entire table or only main columns. In the latter case, only spots' properties and the *Foreground Mean* values of the two channels are exported.

K-Means Clustering

The spots on the scatter plot can be classified into a number of groups according to their quantified values (Figure 23).

- Click the *Analysis > K-Means Clustering* menu.
- Enter the number of clusters (i.e. groups) that you would like.

On the plot, spots close to each other will be grouped into one cluster and highlighted with the cluster's color. At first the table shows the data of all clusters. You can limit it to a cluster using the *Clusters* list-box in the *K-Means Clustering* section. To obtain a different number of clusters, click the *Update Clusters* button. Like with the scatter plot, the table's data can be exported to a CSV file.

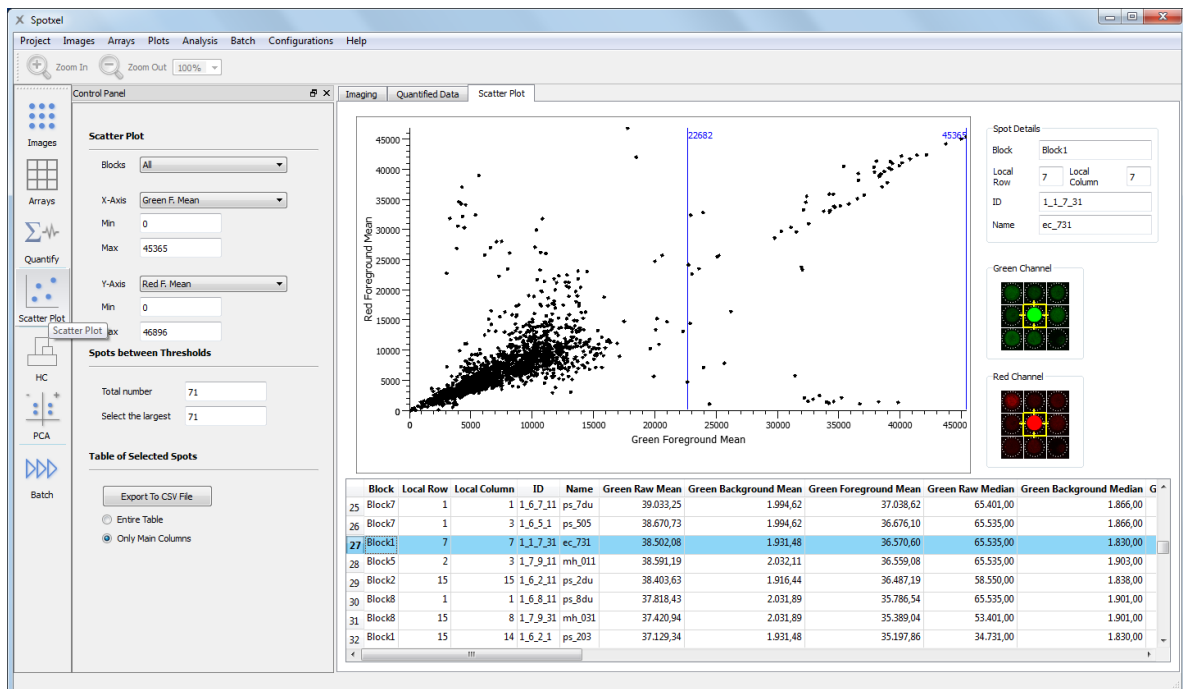


Figure 22: Scatter Plot.

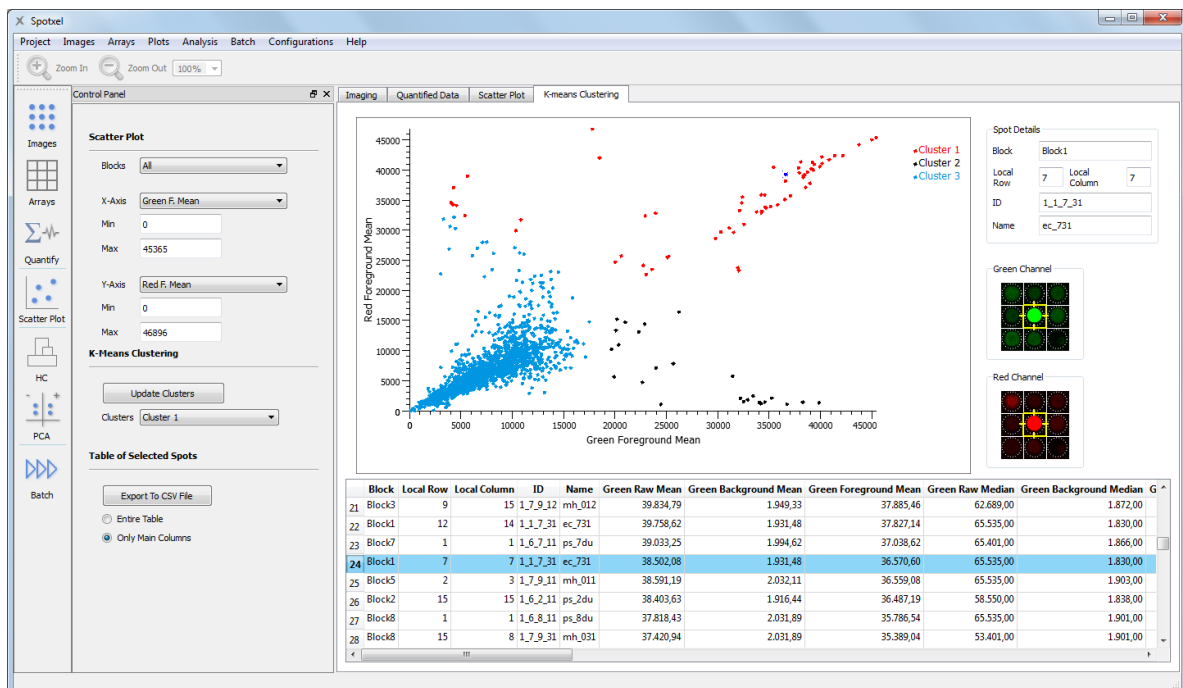


Figure 23: K-Means Clustering.

6 Batch Processing

You can setup a batch to process a number of microarray images automatically. Suppose that the experiment is to screen a protein microarray with k samples. The microarray design is annotated by the so-called *template array*. From the screening result you have k scanned images and would like to quantify their data. To this end, for each scanned image the batch aligns the template array with the image, creates the GAL file that contains the aligned layout, and generates the quantified data.

Click the *Batch* button in the main toolbar to create a batch (Figure 24). In the *Batch* control panel:

- Click the *Add* button and select the microarray images for processing. They will be added to the scheduling table. Use the *Add*, *Remove*, *Up*, and *Down* buttons to modify the table.
- Double-click on the *Template array* edit-box to browse to the template array file.
- Specify the folder to store generated files and the running mode.
- Finally, save the batch to a file using the *Batch > Save Batch* menu. The batch log is created automatically and named after the batch file.

We recommend using a separate folder for each batch to store the batch file and generated data. Since the software uses the dot character (".") for file extensions such as .gal or .csv, please *do not name folders or files used in a batch with dot characters* (except for the file extension) to avoid errors.

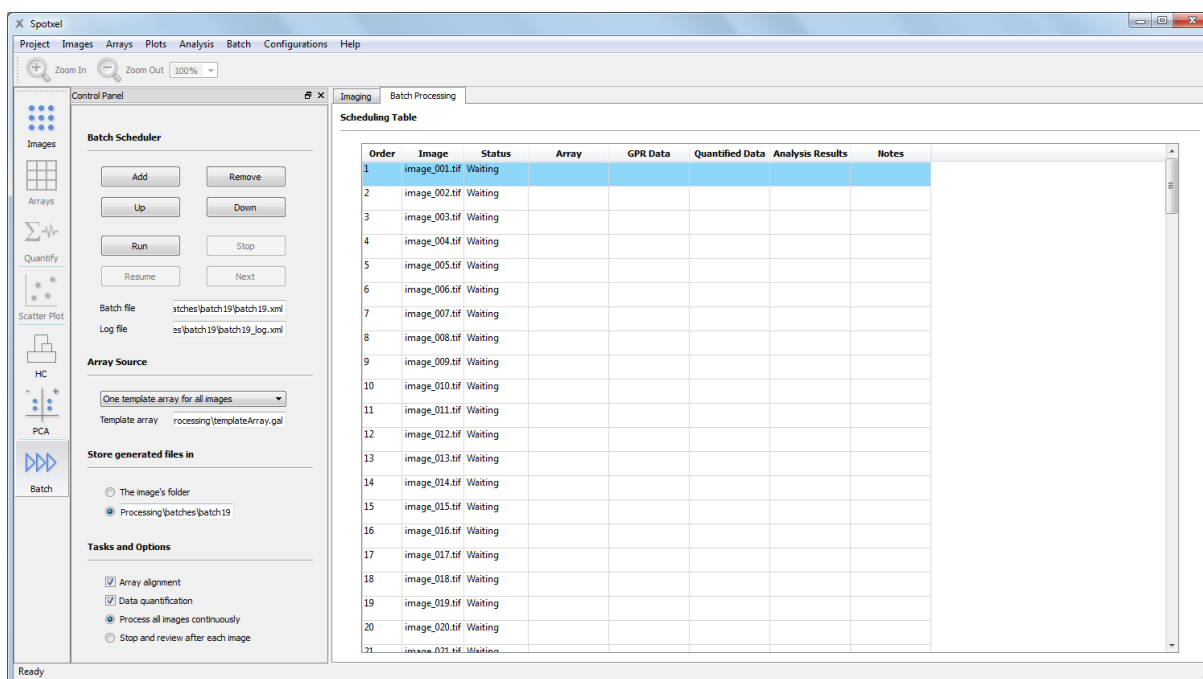


Figure 24: Batch Setup.

After creating the batch, click the *Run* button to execute it (Figure 25). Please note the *running mode*:

- *Process all images continuously*: The batch processes continuously without stopping.
- *Stop and review after each image*: You can view the batch results for one image before proceeding to process the next one.

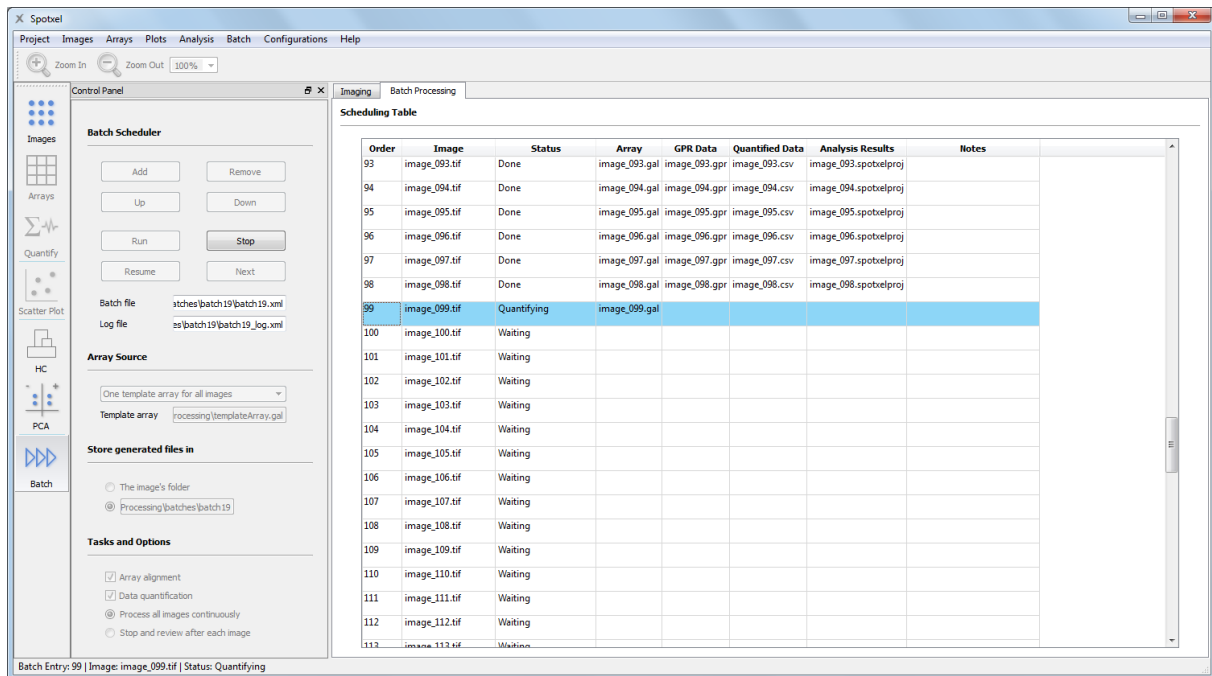


Figure 25: Batch Execution.

Suppose that *sample001.tif* is an image in the batch. The software creates four data files for it:

- *sample001.gal*: the array file whose spot layout is aligned with the image *sample001.tif*,
- *sample001.csv*: a CSV file containing only the quantified data,
- *sample001.gpr*: a file containing quantified data in the GenePix Result (GPR) format, and
- *sample001.spotxproj*: the Spotxel® project containing the analysis data for this image.

7 Data Preprocessing and Normalization

The *Data Preprocessing and Normalization* tool (Figure 26) provides convenient ways to improve the quality of microarray data. *Replicate processing* eliminates duplicates and stabilizes the features' signal value; it consolidates all replicates' signal value into a single value using their mean or median. With the *data filtering* tool you can easily narrow the dataset and focus only on features of interest. *Data normalizations* reduces effects caused by technical variants, thus enables data to be comparable to find the actual relationships between samples and features. You can normalize the

data into standard *Z-Score* values or into ratios to signal values of some *calibration probes*, or measure the screening quality with the *Z-Factor* values.

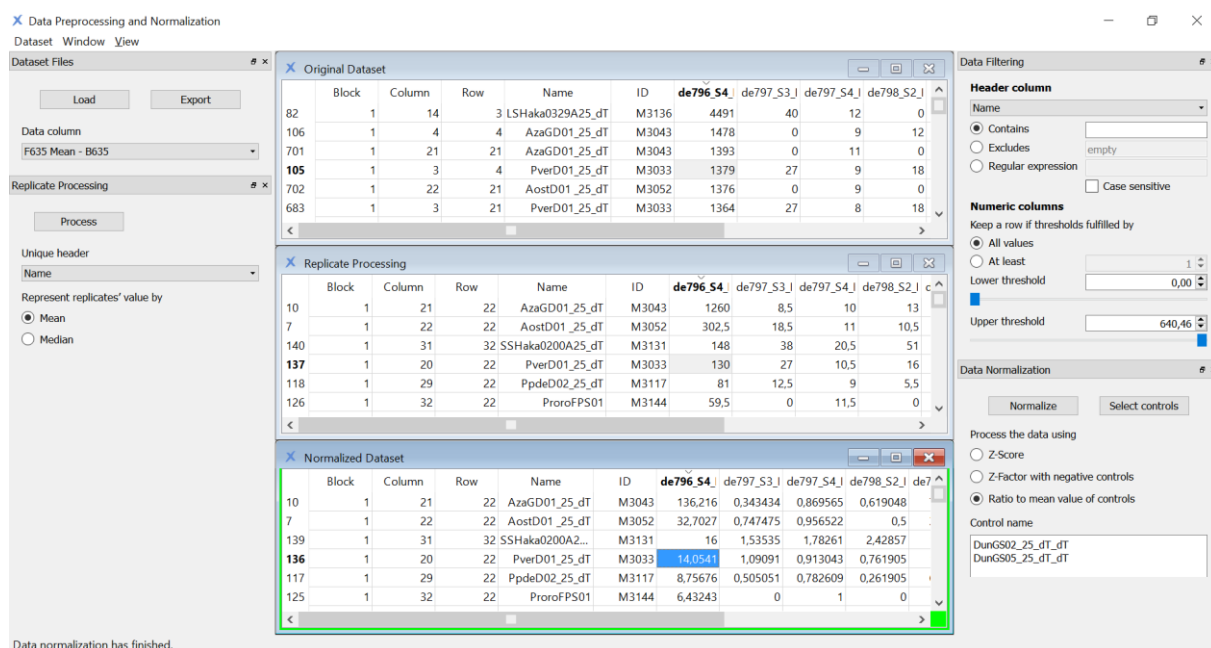


Figure 26: Replicate Processing, Data Filtering, and Data Normalization.

To launch the *Data Preprocessing and Normalization* tool, click the *Norm* button on the main toolbar. Alternatively, you can use the *Analysis > Data Preprocessing and Normalization* menu.

The *Data Preprocessing and Normalization* tool works on datasets which contain the features' signal value. Section 7.1 explains steps to prepare a dataset from different sources. Replicate processing is explained in Section 7.2. Details on data filtering and data normalization are provided in Sections 7.3 and 7.4, respectively.

7.1 Datasets

A dataset can be compiled from a list of Spotxel® projects. Consider the example in Section 6 again, in which the protein microarray is screened with k samples. After running the batch we obtained k Spotxel® project files containing the quantified data. If the dataset is created from these k projects, it can be regarded as the table in Table 3, where V_{1k} is a screening value of *Feature 1* when the microarray is screened against *Sample k* and so on. The screening value can be chosen among the list of quantified values, e.g. *Red Foreground Mean*.

Please note that the first five columns in Table 3 contain the spot's properties specified in the GAL file (i.e., *Block*, *Row*, *Column*, *ID*, and *Name*) of an individual *feature* (a protein in the batch example). For simplicity we only write *Feature 1* instead of its five property values.

Block	Row	Column	ID	Name	Sample 1	Sample 2	...	Sample k
Feature 1					V_{11}	V_{12}	...	V_{1k}
...				
Feature n					V_{n1}	V_{n2}	...	V_{nk}

Table 3: A Sample Dataset.

You can also create a dataset from a list of *GenePix Result (*.gpr)* files. Simply select the .gpr files and then a screening value such as *F635Mean – B635*.

The third way to create a dataset is to prepare a CSV file that has the data format like those in Table 3. The first five columns are *Block*, *Row*, *Column*, *ID*, and *Name*. These *header columns* annotate the features and are usually in the string format. The next columns contain only numeric values; each *numeric column* provides the signal value of the microarray features screened against a sample.

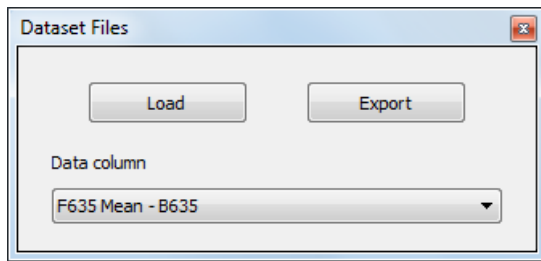


Figure 27: Dataset Files Panel.

To load a dataset from a list of Spotxel project files or .gpr files, in the *Dataset Files* panel (Figure 27 and Figure 26) click on the *Load* button, and select the files. When the files are loaded, specify the screening value in the *Data column* list-box. If the dataset is already prepared as a CSV file then simply load that file. The dataset, with both header columns and

numeric columns, is shown in the table titled *Original Dataset* (Figure 26).

Any dataset (being original, filtered, or normalized) can be exported to a CSV file for further analysis. Simply select by clicking on the corresponding table, which highlight its border in green (such as the *Normalized Dataset* table in Figure 26), and click the *Export* button in the *Dataset Files* panel. If there exist more than two tables of data, you can arrange them using the *Window > Tile* or *Window > Cascade* menu.

7.2 Replicate Processing

Suppose you print each feature (a protein or a gene) in duplicate, triplicates, or more on the microarray for quality and error control. The signal value of a feature's replicates is usually different. In addition, you want only one an instance of the feature in the dataset. This can be done quickly with the *Replicate Processing* tool.

In the *Replicate Processing* panel (Figure 28 and Figure 26); select the header (such as Name, ID, or Peptide) in the *Unique header* list-box. Then click the *Process* button. In each column of the dataset, features having the same Name/ID/Peptide will then be consolidated into one row. The new value will either be the mean or the median of the feature's replicates, as chosen at the *Represent replicates' value* option. A processed dataset is then stored in the *Replicate Processing* table (Figure 26).

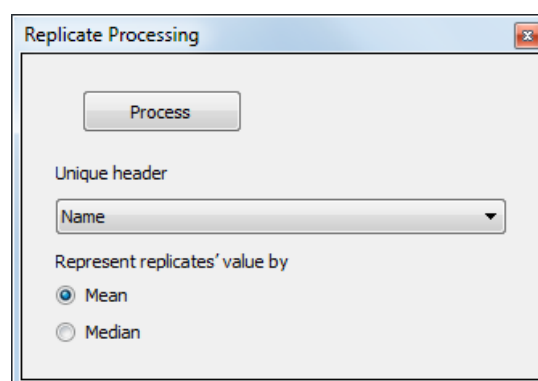


Figure 28: Replicate Processing.

7.3 Data Filtering

You can narrow the currently selected dataset to certain rows, which contain features of interest, by means of header filters. These can be *containing* one or multiple keywords or *excluding* some, or matching a *regular expression* search pattern. In addition, you can select only features whose signal value is within a certain range specified by the *lower* and *upper thresholds*. These tools are provided in the *Data Filtering* panel (Figure 29 and Figure 26).

The header filter is set by first selecting the header column such as Name, ID, or Peptide. You then have the following filtering options:

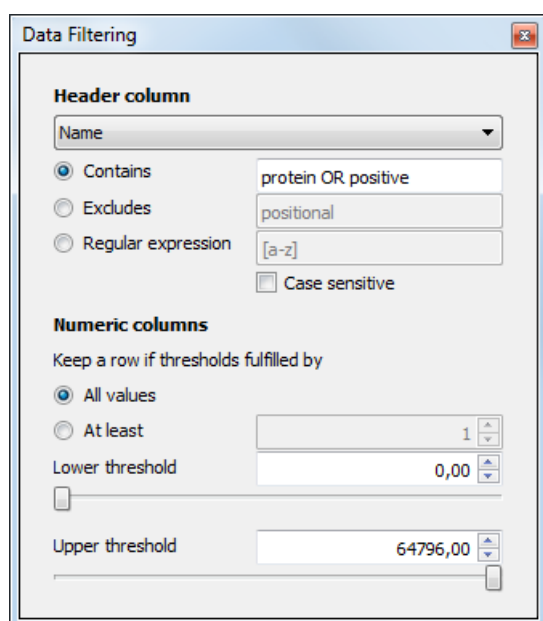


Figure 29: Data Filtering.

- **Contains:** The dataset is narrowed to rows that contain one or more input keywords. A list of keywords must be separate by AND/OR (in the exact uppercase form). For example “protein OR positive” will select rows whose name contain “protein” or “positive”.
- **Exclude:** Rows whose header containing the keywords (separated by OR) are not in the filtered dataset.
- **Regular expression:** This is the most powerful filter based on [regular expression](#). Enter a regular expression search pattern and press Enter to filter the dataset.

The minimum allowed value and the maximum allowed value in the dataset can be set by means of the lower threshold and the upper threshold for the numeric columns. You can condition a row to satisfy all numeric columns, or only a number of them. Please note that a header filter has higher priority than the threshold setting. When you apply a new header filter, the lower and upper threshold values will be reset to the minimum and maximum value of the newly filtered dataset. Starting with these two thresholds you can narrow down the dataset based on the numeric values.

7.4 Data Normalization

Due to technical variations, the absolute signal values obtained directly after quantifying microarray images may not comparably reflect the actual biological change. Data normalization is necessary in these cases to make the data to be comparable again. The *Data Normalization* tool (Figure 30 and Figure 26) provides three methods to normalize the numeric values:

- *Z-Score*: Calculate the relative signal value of a feature as how far, in terms of standard deviations, and in what direction a feature deviates from the central of all features.
- *Z-Factor with negative controls*: Check the screening quality using the standard deviation and the mean value of screening features as well as those of negative controls (Zhang et al., 2000)².
- *Ratio to mean value of controls*: This is useful if the study employs some controls as calibration probes. It calculates the normalized value as the ratio of the feature's signal value to the mean of selected controls' signal value.

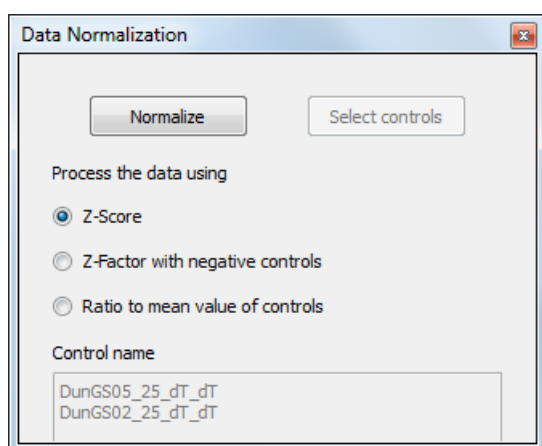


Figure 30: Data Normalization.

If you choose the latter two methods, click on the *Select Controls* button. This shows the *Control Selection* dialog (Figure 31) with a list of all features in the currently selected dataset. There you can choose one or more features as controls for the chosen normalization method.

After choosing a method, click on the *Normalize* button. The normalized dataset, titled *Normalized Dataset*, is shown in the central window (Figure 26).

² Zhang, J-H, Chung T.D.Y., and Oldenburg K.R. (2000) Confirmation of Primary Active Substances from High Throughput Screening of Chemical and Biological Populations: A Statistical Approach and Practical Considerations. J. Com. Chem. 2: 258-265

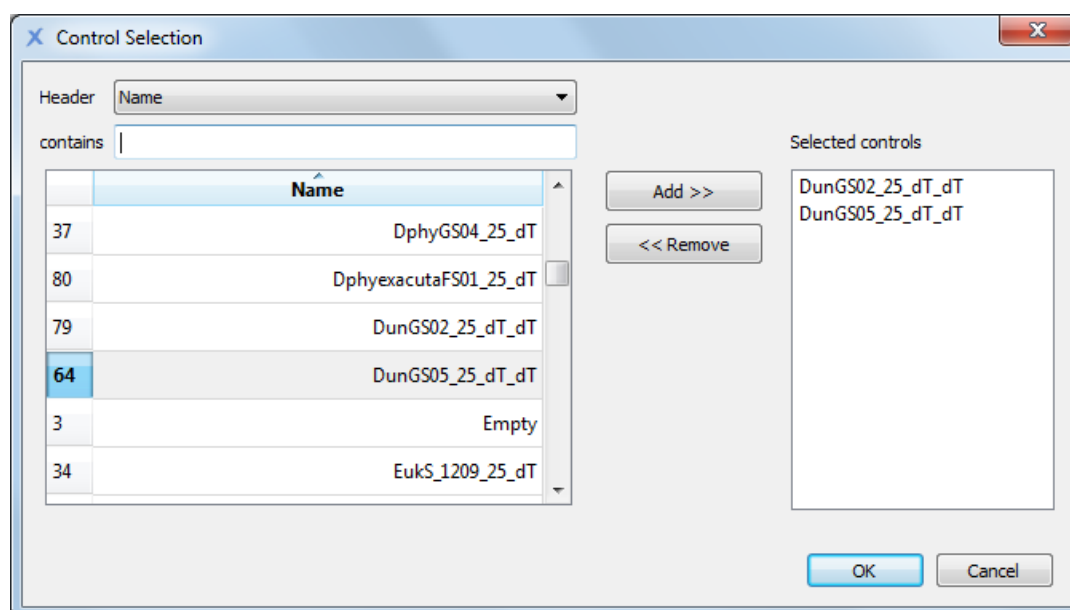


Figure 31: Control Selection for Data Normalization.

8 Data Mining Tools

Data mining tools assist you to find useful information from the microarray study. You can employ *Principal Component Analysis* to discover features and samples that influence the study and then *Hierarchical Clustering Analysis* to find their relationship. The batch processing results, i.e. generated Spotxel® project files, can be used directly for data mining.

Please note that these tools work on datasets. The steps to prepare them from different sources are described in Section 7.1.

8.1 Principal Component Analysis (PCA)

PCA simplifies a complex microarray study to a simpler one with only three samples or features, thus you can easily observe the study's data and its trends. To start with, click the *PCA* button in the main toolbar. In the *PCA* control panel:

- Click the *Load Data* button and select the dataset. Please refer to Section 7.1 for the preparation of the dataset.
- Select a quantified value in the *Data Column* list-box.
- Choose to have the simplified dataset with three either *Features* or *Samples*.
- Click the *Start Analysis* button.

Block	Row	Column	ID	Name	Sample x	Sample y	Sample z
Feature 1					V_{1x}	V_{1y}	V_{1z}
...				
Feature n					V_{nx}	V_{ny}	V_{nz}

Table 4: The Simplified Dataset.

Suppose that that you chose the *Samples* option. Take the screening in the previous section again as example. As shown in Table 3, the original dataset represents the features' screening value against k samples, where k is much larger than 3. PCA will simplify it to a dataset with only 3 samples, as illustrated in Table 4.

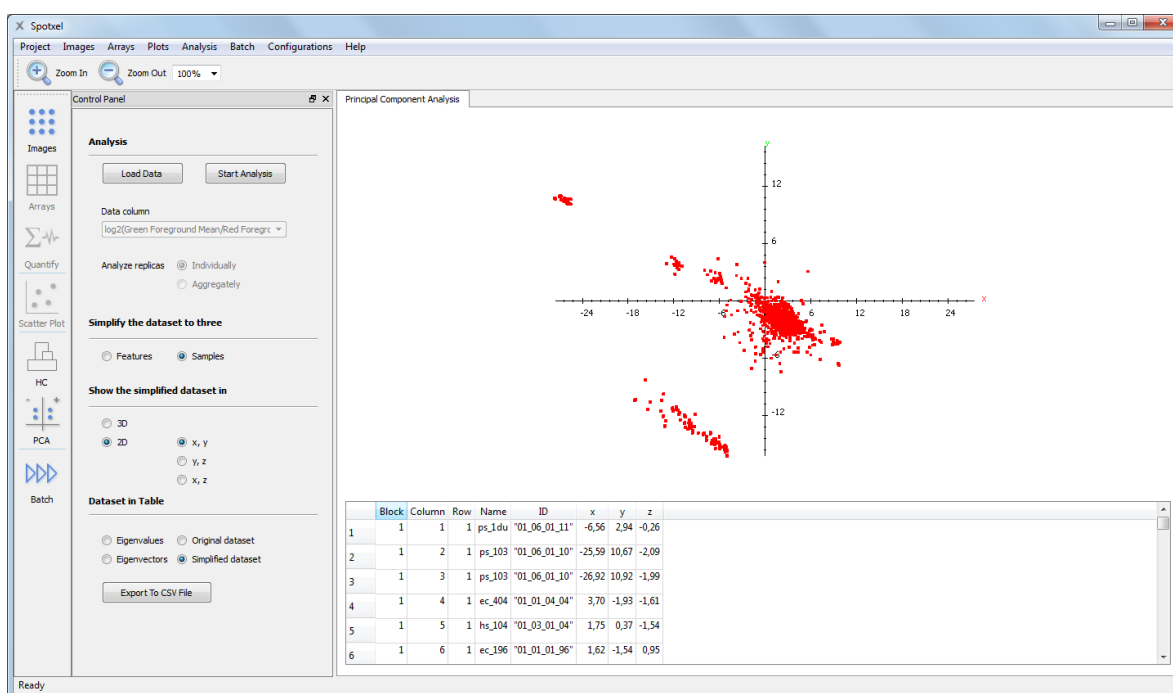


Figure 32: Principal Component Analysis in 2D View.

The PCA chart (Figure 32) then depicts the features according to their values in the simplified dataset, whose data is shown in the table below the chart. You can select 2D or 3D view. By observing the charts you can find the trends of the data. For example, features (proteins in the example) that have common characteristics locate near each other on the chart. On the other hand, those that are distinct are far from the others. The simplified dataset can be exported to a CSV file for further analysis.

Similarly, you can discover such information about the samples by choosing the *Features* option before starting the analysis. By combining the results of these two analyses, you may be able to discover features (proteins in the example) and samples that influence the variance of the study.

8.2 Hierarchical Clustering Analysis (HCA)

You can group features or samples that are related using HCA. The relationship can be e.g. having similar effect in the study, represented by close screening values. Click the *HC* button in the main toolbar to setup the analysis.

- Click the *Load Data* button and select the dataset. Please refer to Section 7.1 for the preparation of the dataset.
- Select a quantified value in the *Data Column* list-box.
- Choose to construct the clustering tree for *features*, or *samples*, or both.
- Select the *distance metric* and the *type of linkage*. You can keep the default options.
- Click the *Start Analysis* button.

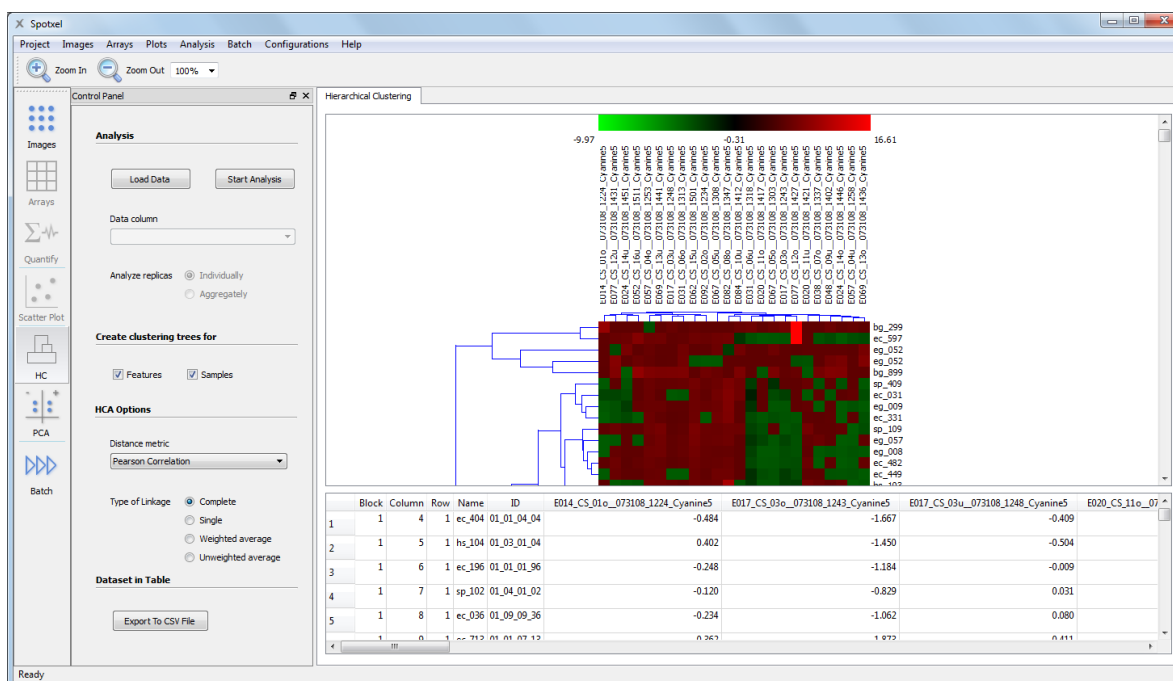


Figure 33: Hierarchical Clustering Analysis.

The clustering tree(s) are then constructed (Figure 33). Two features (proteins in the batch example), which are considered being *related*, are grouped into one *cluster*. Their *relationship* is represented by a line connecting them. A cluster might be related with a feature or another cluster. The relationship between samples and clusters of samples are represented similarly.

The values in the dataset, each representing the screening value of a feature with a sample, are graphically represented by means of a heat map. You can save the clustering trees with the heat map to an image file using the *Export to Image* context menu.

9 Product Activation

The product activation requires an *internet connection*. You need to have a *serial number* obtained from the software provider or its distributors.

- (1) In the *Evaluation Time Has Expired* dialog, click the *Next* button.

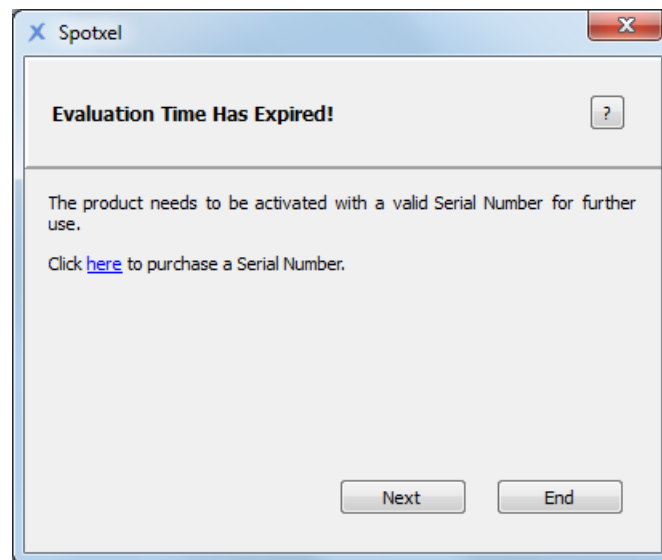


Figure 34: Starting the Product Activation.

- (2) Enter the serial number and the licensee information in the *Product Activation* dialog. Click *Next*.

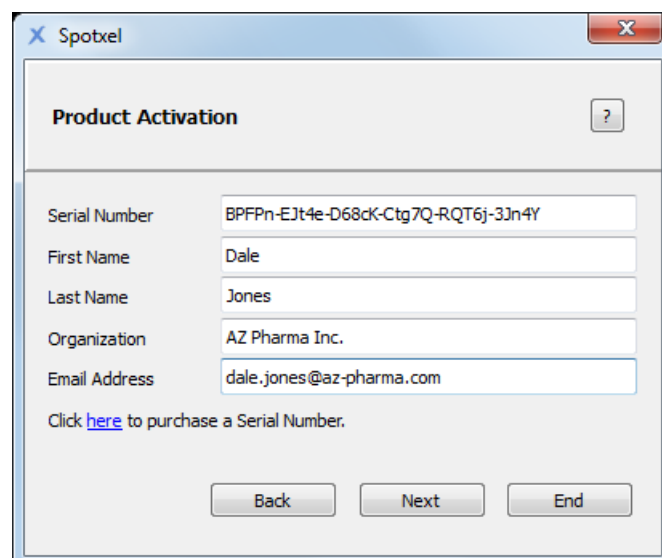


Figure 35: Entering the Licensee Information.

- (3) If the internet connection is ready, click the *Activate* button and wait for the activation to finish.

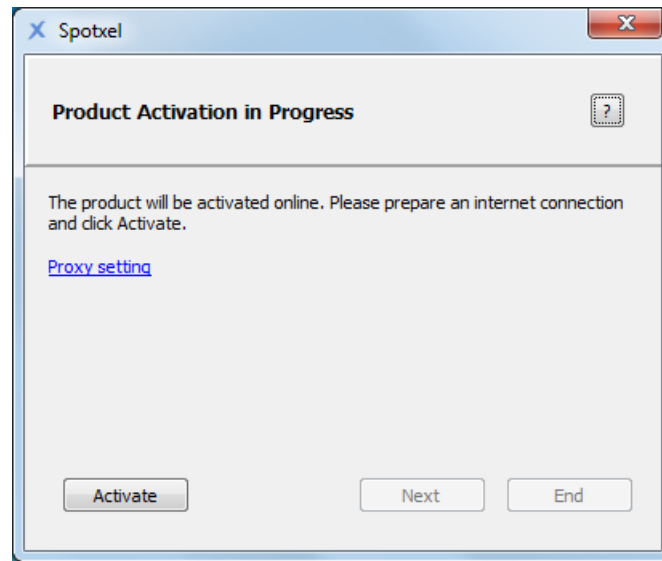


Figure 36: Product Activation in Progress.

Please check the internet connection in the case the software could not reach the activation server. If your system uses a proxy server to connect, specify it using the *Proxy setting* link. Otherwise, please contact the software provider for support.

- (4) A completion message is shown when the product is successfully activated. Click *Next* to use the software immediately or *End* to use it later.

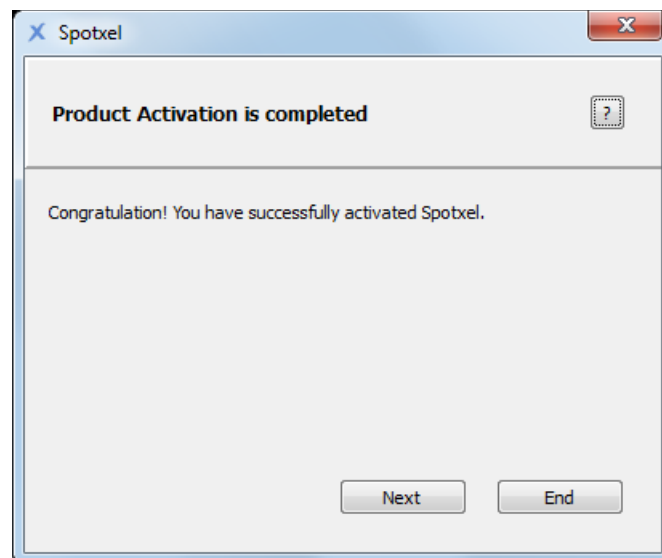


Figure 37: Completion of the Product Activation.

Please note that the license can be reviewed, or renewed in the case of a time-limited license, by clicking on the *Help* menu and choosing *License Information*.

10 End-User License Agreement

SPOTXEL IS THE PROPERTY OF SICASYS SOFTWARE GMBH ("SICASYS"). THE USE OF THIS SOFTWARE IS GOVERNED BY THE TERMS AND CONDITIONS OF THE END-USER LICENSE AGREEMENT ("AGREEMENT") SET FORTH BELOW. THE TERM "SOFTWARE" ALSO INCLUDES RELATED DOCUMENTATION (WHETHER IN PRINT OR ELECTRONIC FORM) AND ANY UPDATES OR UPGRADES OF THE SOFTWARE PROVIDED BY SICASYS.

BY INSTALLING THE SOFTWARE, AND/OR BY USING THE SOFTWARE, YOU AGREE TO BE BOUND BY THE TERMS AND CONDITIONS OF THIS END-USER LICENSE AGREEMENT.

License. SICASYS grants to you ("USER") a non-exclusive, non-transferable license to use the SOFTWARE on the number of computers stated in the license contract. A separate license is required for use on any additional computer.

Copy Restriction. The SOFTWARE may not be copied either in full or part by USER, with the exception of making copy for security or backup purpose. Copies must include all copyright and trademark notices.

Use Restrictions. This SOFTWARE is licensed to USER for internal use only. USER shall not (and shall not allow any third party to):

- (i) decompile, disassemble, reverse engineer or attempt to reconstruct, identify or discover any source code, underlying ideas, underlying user interface techniques or algorithms of the SOFTWARE by any means whatever, or disclose any of the foregoing;
- (ii) modify, incorporate into or with other software, or create a derivative work of any part of the SOFTWARE;
- (iii) attempt to circumvent any user limits, or other license, timing or use restrictions that are built into the SOFTWARE.

Ownership of the SOFTWARE. SICASYS retains all titles of ownership, all ownership rights, and all intellectual property rights of the SOFTWARE. SICASYS reserves all rights not expressly granted to USER.

Termination. SICASYS may terminate this Agreement immediately if USER breaches any provision. Upon notice of termination by SICASYS, all rights granted to USER under this Agreement will immediately terminate, and USER shall cease using the SOFTWARE and return or destroy all copies of the SOFTWARE.

Limited Warranty and Disclaimer. USER is aware of the fact that technical errors in the program in accordance with the accompanying documentation cannot be excluded. If USER claims deviations in the program from the specification/description, USER has the right to request a fix, patch, work-around, or replacement of the SOFTWARE that does not meet such limited warranty. If a rectification of the errors is not possible or if such rectification is not successful, USER has the right to request a rescission of the contract, where USER must destroy all copies of the SOFTWARE.

EXCEPT AS EXPRESSLY SET FORTH ABOVE, NO OTHER WARRANTIES OR CONDITIONS, EITHER EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, ARE MADE BY SICASYS WITH RESPECT TO THE SOFTWARE AND THE ACCOMPANYING DOCUMENTATION, AND SICASYS EXPRESSLY DISCLAIMS ALL WARRANTIES AND CONDITIONS NOT EXPRESSLY STATED HEREIN, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, NONINFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE. SICASYS DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE SOFTWARE WILL MEET USER'S REQUIREMENTS, BE UNINTERRUPTED OR ERROR FREE, OR THAT ALL DEFECTS IN THE PROGRAM WILL BE CORRECTED. USER ASSUMES THE ENTIRE RISK AS TO THE RESULTS AND PERFORMANCE OF THE SOFTWARE.

Limitation of Liability. IN NO EVENT SHALL SICASYS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL OR INCIDENTAL DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTIONS, LOSS OF BUSINESS INFORMATION, OR OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE, EVEN IF SICASYS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. FURTHER, IN NO EVENT SHALL SICASYS BE LIABLE FOR ANY DIRECT DAMAGES ARISING OUT OF USER'S USE OF THE SOFTWARE. IN NO EVENT WILL SICASYS BE LIABLE TO USER FOR DAMAGES IN AN AMOUNT GREATER THAN THE FEES PAID FOR THE USE OF THE SOFTWARE.

Intellectual Property Right Infringement. If a claim alleging infringement of an intellectual property right arises concerning the SOFTWARE (including but not limited to patent, trade secret, copyright or trademark rights), SICASYS in its sole discretion may elect to defend or settle such claim, and/or terminate this Agreement and all rights to use the SOFTWARE, and require the return or destruction of the SOFTWARE, with a refund of the fees paid for use of the SOFTWARE less a reasonable allowance for use and shipping.

Miscellaneous. This Agreement is the entire agreement between USER and SICASYS with respect to the SOFTWARE, and supersedes any previous oral or written communications or documents (including, if USER is obtaining an update, any agreement that may have been included with the initial version of the Software). This Agreement is governed by the laws of Germany. If any provision, or portion thereof, of this Agreement is found to be invalid or unenforceable, it will be enforced to the extent permissible and the remainder of this Agreement will remain in full force and effect. Failure to prosecute a party's rights with respect to a default hereunder will not constitute a waiver of the right to enforce rights with respect to the same or any other breach.

11 Index

A

activate · 1, 30, 31
 analysis of microarray data · 4
 array · 3
 array alignment · 6
 alignment control · 9
 automatically · 6
 manually · 6
 offset of adjacent blocks · 10
 options · 9
 rotation detection · 10
 signal · 9

B

background control · 18
 association · 18
 background correction · 17, 18
 background control · 18
 for a block · 18
 global background value · 18
 Local method · 14, 18
 block level · 18
 global level · 18
 spot level · 18
 background value · 4, 12
 batch · 4, 21
 create · 21
 execute · 22
 template array · 21
 block · 3
 properties · 5
 brightness · 12

C

canvas · 2, 5
 canvas toolbar · 2
 cluster · 29
 contrast · 12
 control panel · 2

D

data filtering · 22, 25
 header filter · 25
 lower threshold · 26

 upper threshold · 26
 data mining · 4
 dataset · 23, 27, 29
 Hierarchical Clustering Analysis · 29
 K-Means clustering · 19
 Principal Component Analysis · 27
 data normalization · 22, 26
 ratio to mean value of controls · 26
 Z-Factor · 26
 Z-Score · 26

F

file
 CSV · 24
 GAL · 3, 6, 22
 GenePix Result (*.gpr) · 24
 microarray image · 6
 Spotxel® project (*.spotxelproj) · 6, 23
 TIFF (*.tiff, *.tif) · 5
 flag · 12
 Flex-Spot · 13
 detected spot border · 13, 17
 smallest spot size (%) · 17
 foreground value · 4, 12

H

hardware requirement · 1
 HCA · 29
 clustering tree · 29
 heat map · 29
 Hierarchical Clustering Analysis · 29

I

Installation · 1
 Mac OS X platforms · 1
 Windows Platforms · 1
 inversion · 8

K

K-Means clustering · 19

L

license

buy · 2

End-User License Agreement · 32

M

main toolbar · 2, 3

menu · 2

N

noise

background · 15

foreground · 16

process noise at quantification · 16

noise filtering · 8

P

PCA · 27

original dataset · 28

simplified dataset · 28

Principal Component Analysis · 27

Q

quantification · 4, 11, 13, 17, 18

change of intensity values · 13

quantified data · 12

export to CSV file · 12

table · 2

R

raw value · 4, 12

replicate processing · 22, 24

rotate

array (blocks) · 8, 9

image · 8, 9

S

scatter plot · 19

serial number · 30

spot

border · 3

diameter · 3

properties · 5, 12, 19

smallest spot size · 17

spot detection method · 13

Fixed-Spot · 13

Flex-Spot · 13

Spot Image widget · 2, 12, 19

spotted region · 3, 13

Z

zoom · 5