

**Aracne**  
Version 3.12

**A Software Tool for Thread Counting in X-rays of  
Fabrics**

---



**Aracne**  
Version 3.12

# **A Software Tool for Thread Counting in X-rays of Fabrics**

---

**Juan José Murillo Fuentes, Irene Fondón García, Marta  
Ternero Gutiérrez,  
Lucía Córdoba-Saborido and Pablo Aguilera Bonet**

Teoría de la Señal y Comunicaciones  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla



© 2018. Juan José Murillo Fuentes, Irene Fondón García, Marta Ternero Gutiérrez,  
Lucía Córdoba-Saborido and Pablo Aguilera Bonet

© Grupo de Aprendizaje, Procesado de señal y Comunicaciones. Departamento de Teoría de la Señal y  
Comunicaciones.  
Universidad de Sevilla

email: murillo@us.es  
Sevilla, 2018

Compiled August 27, 2018 at 2:41 p.m.

**The (re)distribution of this software is not permitted. The software is licensed to be used with no restrictions. However, if you or your institution find it useful, we encourage you to write us with your feedback. Besides, we kindly ask to cooperate in this project with funding, that will be invested into further improving the usability of this software and developing new tools. You may contact us writing to [murillo@us.es](mailto:murillo@us.es).**

Diseño de maquetación: F. Javier Payán Somet © 2018.

# Preface

---

**T**HE Aracne software was mainly designed by Juan José Murillo-Fuentes, Irene Fondón-García, Marta Ternero-Gutiérrez, Lucía Córdoba-Saborido and Pablo Aguilera-Bonet. It was born as a collaboration with Museo Nacional del Prado to develop a tool to count threads in X-rays of canvases. We are deeply grateful to Laura Alba, for her useful comments, suggestions, test and supervision of the Aracne software.

The main objective was to provide a tool to count vertical and horizontal threads along large areas in a X-ray image. We have developed a software not just fulfilling this request, but with the following additional features:

- Thread counting along the whole image, for every spot in the canvas.
- Analysis of the angle deviation of the threads from the horizontal and vertical axes.
- Analysis of the power spectrum density.

This software is a beta version, meaning that we are still working on its improvement. Any feedback or support will be acknowledged, please write us to [murillo@us.es](mailto:murillo@us.es). If you wish to support our work or are interested in any further improvement, do not hesitate to contact us.



# Contents

---

<i>Preface</i>	I
<i>Contents</i>	III
<b>1. Installation</b>	<b>1</b>
1.1. First installation	1
1.2. Updates	1
1.3. Uninstall	2
<b>2. A quick start</b>	<b>3</b>
2.1. Opening the tool	3
2.2. Load an image	4
2.3. Selecting an area of the image	4
2.4. Running the thread counting algorithm	6
2.4.1. Results: Thread counting	6
2.4.2. Results: Power spectral density	8
<b>3. Visualization options</b>	<b>11</b>
3.1. Visualization of the results	11
3.1.1. Overlapping results and original image	11
3.1.2. Setting histograms and angle deviation limits	11
3.2. Visualization of the original image	12
3.2.1. Invert gray levels	12
3.2.2. Grid	12
3.2.3. Image properties	13
3.2.4. Zoom	13
<b>4. Saving results, projects</b>	<b>15</b>
4.1. A project	15
4.1.1. Saving a project	15
4.1.2. Opening an existing project	15

4.2. Saving results images	16
<b>5. Parameters</b>	<b>19</b>
5.1. Algorithm overview	19
5.2. Possible and Default values	20
5.3. Save and Load Parameters	21
5.4. Preprocessing	21
<b>Appendix A. Image Format</b>	<b>23</b>



# 1 Installation

---

## 1.1 First installation

**T**O run Aracne it is necessary to have:

- A simple executable file, a AracneV3.X.exe in Microsoft Windows or AracneV3.X.app in Mac OS.
- Installed the MATLAB<sup>®</sup> Compiler Runtime MCR V9.4.

Accordingly the installations steps are as follows:

1. Install the Compiler Runtime MCR V9.4 from MATLAB<sup>®</sup> at  
<https://es.mathworks.com/products/compiler/matlab-runtime.html>.

**Important:** Make sure that the version of the MCR software is 9.4 for the corresponding operative system in which Aracne is wished to run.

2. Copy the file AracneVX.exe (where X reads for the number version of the software), or AracneVX.app if using Mac OS, in any folder, e.g. in the applications folder.

Then you can run AracneVX by doubly clicking on it. In Mac OS, the first time may be needed to run it by using the menu options of the file.

## 1.2 Updates

If you want to update the software with a new version you just copy the file AracneVY.exe, or AracneVY.app if using Mac OS, in any folder. Then check if the version of the MCR is V9.4. If not, you will need to install a new version of the MCR.

## 1.3 Uninstall

To uninstall, simply uninstall MATLAB<sup>®</sup> MCR and delete the file AracneVX.exe, or AracneVX.app if using Mac OS.

## 2 A quick start

---

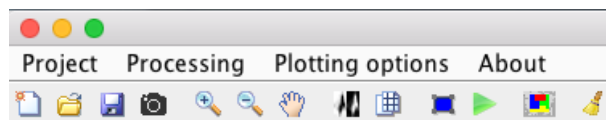
A complete processing of an image involves the following steps:

1. loading the image,
2. selecting an area or the whole image to process,
3. setting the parameters,
4. running the thread counting algorithm,
5. visualizing the results and
6. saving them as a project and/or images.

In this chapter we go over these steps in the simplest way by just loading an image, selecting the whole of it, using the default parameters, running the algorithm and visualizing the results. Later, in the following chapters, we go over these steps in a deeper manner to fully explain, among other details, the meaning of the parameters and the visualization options.


### 2.1 Opening the tool

If correctly installed, after clicking the application icon we should head into the Aracne interface, where we distinguish two main areas, being the menu area and the icon area (Figure 2.1), where some shortcuts to the menu options are set.



**Figure 2.1** Menu and toolbars.

## 2.2 Load an image

First step is loading an image. This can be performed by either clicking on the first icon from the left, , either typing CTRL + N (cmd + N in Mac OS), or by using the Project menu > New image option.

The images allowed are .tif format, in greyscale, with image size, in pixels, and resolution value and resolution unit (either in inches or centimeters). Please, make sure your image meet this requirements, otherwise it will not be loaded, or sizes will be loaded improperly. In Appendix A it has been included an example of the format of an image with compliance parameters.

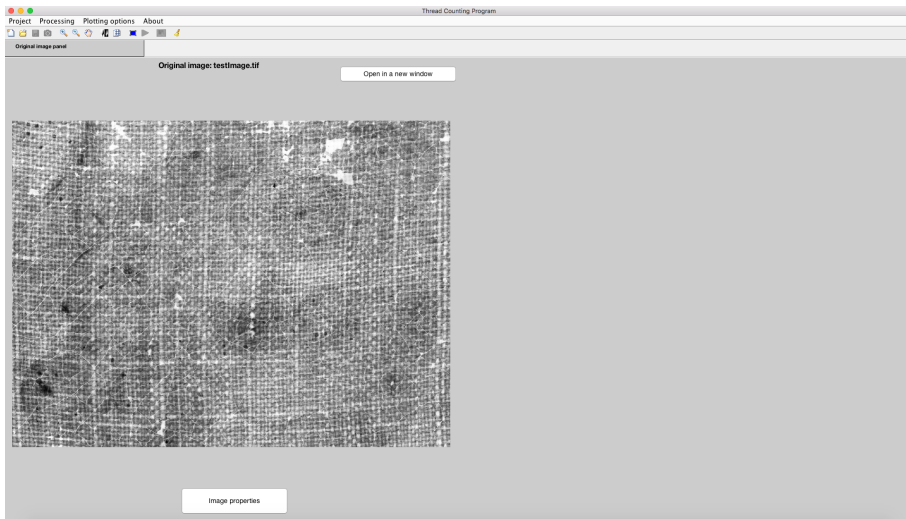



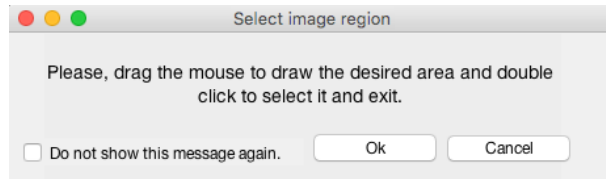
Figure 2.2 Loaded image.

The view of Figure 2.2 is the result of loading a sample image, which is shown in a tab named 'Original image panel'. Once an image is successfully loaded some new options become available, such as zoom tools or a gray levels inversion. These functions and more will however be explained with more detail in .

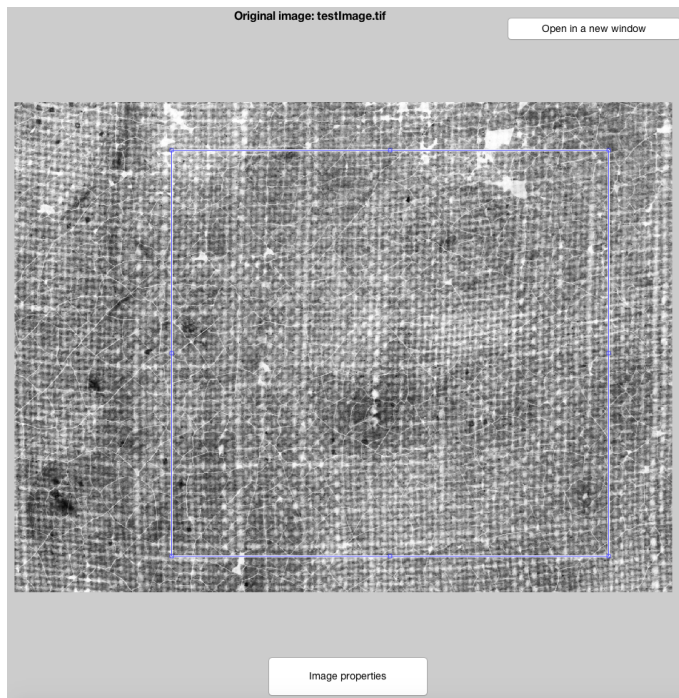
## 2.3 Selecting an area of the image

With Aracne one may study the whole loaded image or just select an area. In this chapter we will process an area of the loaded image by clicking on the icon , by going to Processing > Processing Region > Select image region, or equivalently by pressing CTRL + R (cmd + R in Mac OS).

A new window will show up where the instructions to select a region are explained (Figure 2.3). This window will show up every time unless the 'Do not show this message again' checkbox is selected.



**Figure 2.3** Instructions window for the selection of a image region. This message will pop up everytime Aracne is started.

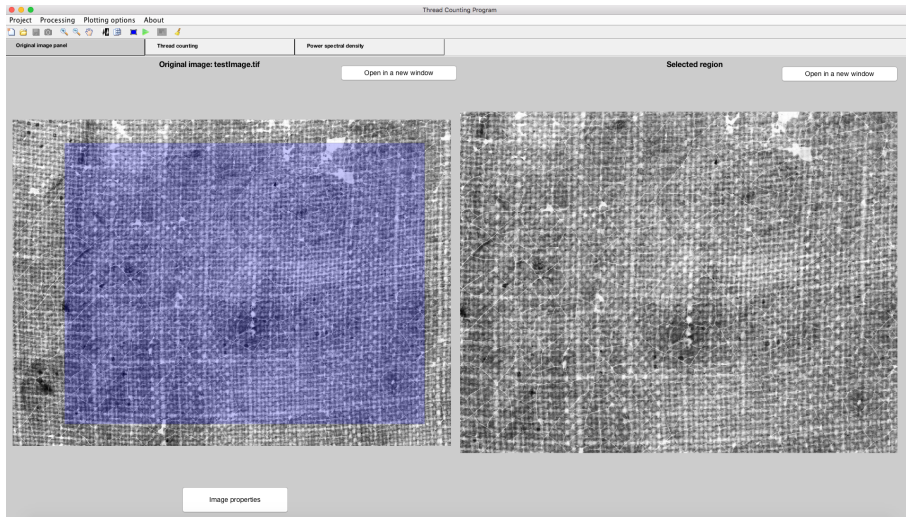


**Figure 2.4** Window to select the to-be processed selected area of the image.

By clicking and dragging the pointer it is possible to draw a rectangle, which can be redefined by selecting its sides or corners and moving them as wished (Figure 2.4). A double click will select the area enclosed by the blue rectangle, and it will be displayed on the right side of the software window, as it can be seen on Figure 2.5. This is the area to be processed.

The software allows to repeat this process to select another area as many times as wanted.

In addition, the full image can be selected and processed if preferred. In this case go to Processing > Processing Region > Select whole region, or press CTRL + W (cmd + W). The full image will be overlapped with the blue selected area, and ready to be processed.



**Figure 2.5** To-be processed area of the image.

## 2.4 Running the thread counting algorithm

Once the user is satisfied with the selected region, the software is ready to start the counting process. To run the algorithm it is possible to proceed just clicking on the play icon, ▶, going to Processing > Begin, or pressing CTRL + B (cmd + B). This will start the algorithm, showing a progressing bar that includes a button to cancel the counting if needed. Although in this section the main procedure is the simplest one, it is possible to set some parameters involving in the process, the reader is referred to Chapter 5 for further specific information.

When finished, two new tabs similar to the 'Original image panel' appear, named 'Thread counting' and 'Power spectral density'.

### 2.4.1 Results: Thread counting

When the thread counting process is finished, a new 'Thread counting' tab is automatically selected, where there are three main distinguishable sections.

The thread counting results are shown in Figure 2.6, which contains four available subtabs. The first two tabs correspond to the thread counting for the vertical and horizontal threads, while the latter two show the thread angle deviation with respect to the horizontal and vertical directions.

Each pixel colour is the result of the counting within a square window (whose default size side is 2cm), and a colour legend helps translating the colour image into values of threads per cm, indicating the result along the selected area.

In the case of the deviation angle, a positive value traduces in a clockwise deviation, while a negative results means a counter clockwise deviation.

The user might find useful the overlap option in order to better observe the results by

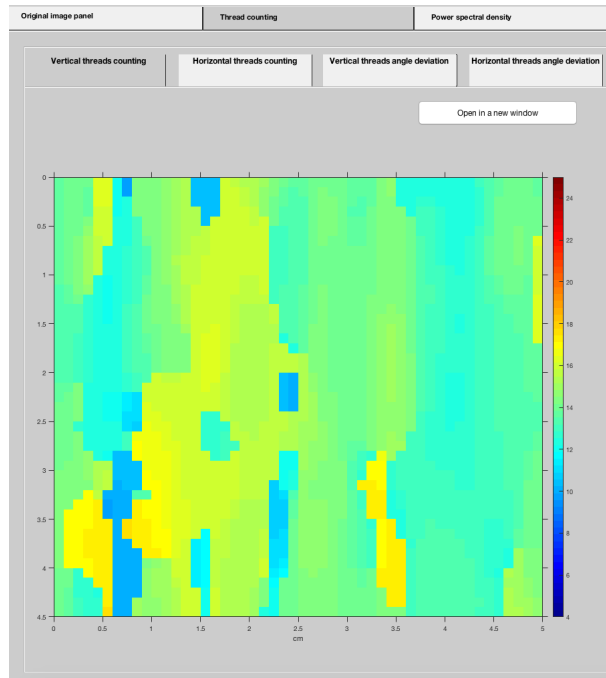


Figure 2.6 Result of the threads counting.

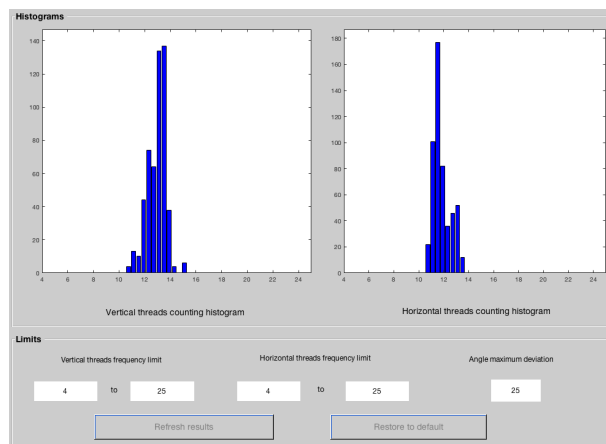


Figure 2.7 Histograms of the results yielded by the thread counting.


locating every color pixel within the selected area. This option is further described in Subsection 3.1.1, and it is activated by clicking on .

Figure 2.7 contains the resulting histograms for the thread counting, both for the vertical and the horizontal threads. These just represent the number of times a thread counting value is found along the processed image. Below these histograms the user can find some

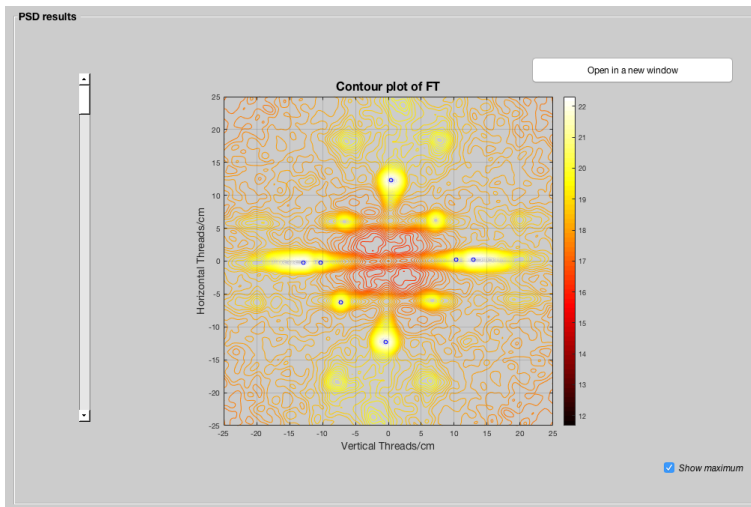
statistical values for the thread counting, the mean, mode and statistical deviation for both the vertical and horizontal results.

It is possible to manipulate the frequency limits of the counting, which will affect the histograms and range of colours of the thread counting results. In addition, a value for the maximum deviation angle can be chosen by the user.

### 2.4.2 Results: Power spectral density

This tab provides the user with information about the frequency-relative calculus of the processing.

Here, there can be found graphic results of the power spectral density (PSD) such as level curves and a surface plot, as well as numerical results about the power spectral density maxima.



**Figure 2.8** Power spectral density.

In Figure 2.8 it is shown a representation of the level curves of the PSD. The user is allowed to interact with the slider on the left of the plot in order to investigate previous stages of the calculation.

Also, the maxima of the plot (which also appear in Figure 2.9) are visible.

The user can set the number of maxima to find (Figure 2.10) as well as select a value of the numerical result's table (by simply clicking), which will turn green in order to be identified. Maxima will be searched within the frequency limits that are set in the 'Plotting options' section, below the mesh plot of the PSD.



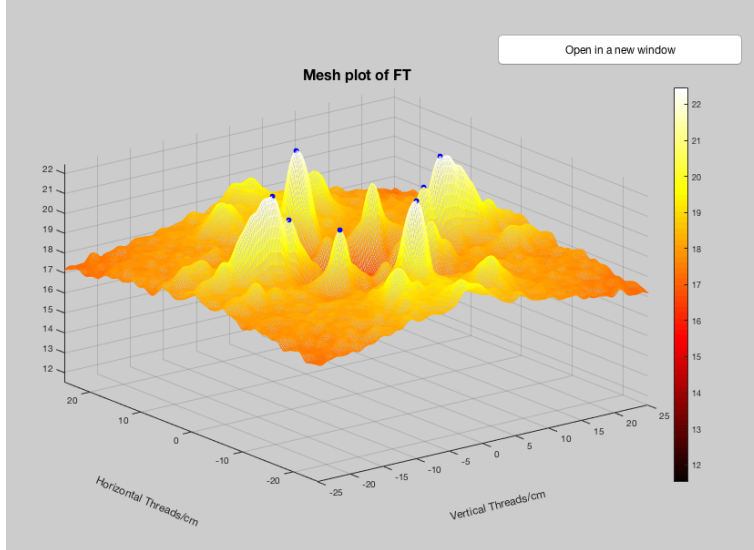


Figure 2.9 Power spectral density.

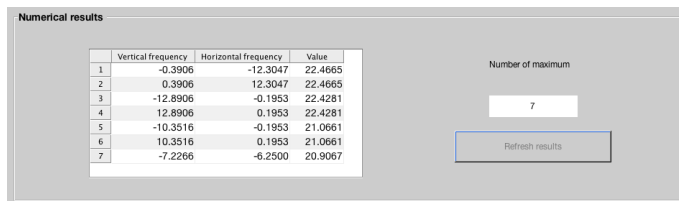


Figure 2.10 Power spectral density.




## 3 Visualization options

---

**T**HERE are some options that help visualizing the results and the original loaded image. In this section some of them are described.

### 3.1 Visualization of the results

#### 3.1.1 Overlapping results and original image

In order to locate the results of the counting within the selected area of the canvas, the user may activate the overlap option. This action becomes possible after running the thread counting algorithm, it is activated by just clicking on the icon  or going to Plotting options > Overlap results with processing area. In Figure 3.1, it is included an example of this superposition, where the color image with the thread counting and the angle deviation results are plotted with transparency in front of the selected area of the original image. In these new images, every color pixel is surrounded by the area used to compute the value it encodes.

#### 3.1.2 Setting histograms and angle deviation limits

Below the histograms area, one can find some boxes containing the limits for its axes. In the example from Figure 2.7, the histograms could be better observed if these limits were set to a 10 - 15 frequency range for the vertical and horizontal threads. Once these limits are introduced, histograms can be plotted again by clicking on the 'Refresh histogram results' button. Note that the 'Numerical results' section (located under the histograms) will be recomputed for this range. Besides, an specific angle deviation maximum value can be selected (while it is chosen as a lower value than the maximum angle selected in the parameter's configuration for the counting, see Chapter 5), though this will not affect the histograms since they are only related to the vertical and horizontal threads frequencies, but it will have some relevance in the third and fourth subtabs.

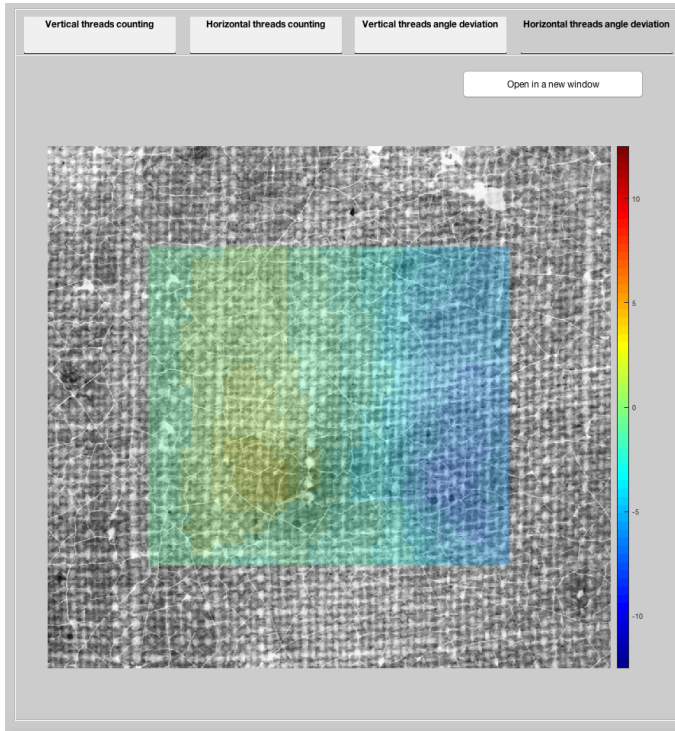



Figure 3.1 Overlap of the results with the original image.


## 3.2 Visualization of the original image

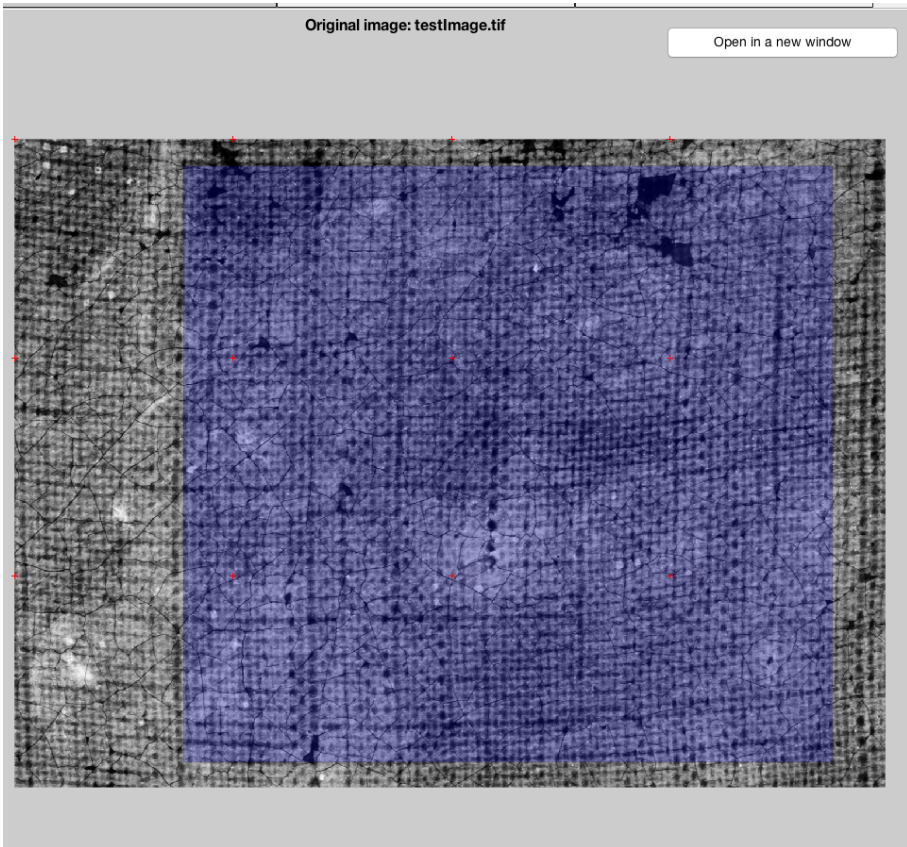
Some tools have been included to improve the user's experience when dealing with the original image, as follows.

### 3.2.1 Invert gray levels

Although Aracne automatically inverts the input images by default, sometimes it can be preferred to use the original image with the intensity levels inverted. To accomplish this the user can either just click on the icon , press CTRL + I (cmd + I), or go to Plotting options > Invert gray levels.

### 3.2.2 Grid

A  $2 \times 2$  cm grid can be overlapped to the original and selected area images by just clicking on the icon , pressing CTRL + G (cmd + G) or going to Plotting options > Add a grid. The result is an squared array of red crosses every 2 cm, allowing for instance a counting by visual inspection. In Figure 3.2 it can be observed the result, with red crosses arranged in a grid on the image.







**Figure 3.2** Original image and its selected region after inverting gray levels and activating the grid option.

### 3.2.3 Image properties

The properties of the loaded image can be observed by clicking on the button 'Image properties', below it.

### 3.2.4 Zoom

All images and plots of the software can be zoomed in and out with the zoom tools by clicking in the icons  and  and then in the images. They can also be opened into a new window if the user is interested in a more detailed view by simply pressing the 'Open in a new window' option of the corresponding plot or image.

There are some options which may be useful in some of these new windows, such as the 'Data cursor' option () in the contour and mesh plots, and the 'Rotate' tool () in the 3D mesh plot.



## 4 Saving results, projects


---

**O**NCE an image has been loaded and the thread counting algorithm has been run, one may wish to save both the results and the configuration used. This can be easily done by saving the so-called project.


### 4.1 A project

When an image is loaded to be processed, the program generates a project, which is an environment where it is included the path to the image and the parameters used to run the thread counting algorithm. Once run, this project can be saved, and then reloaded to visualize the results or events to modify and re-run it.


#### 4.1.1 Saving a project

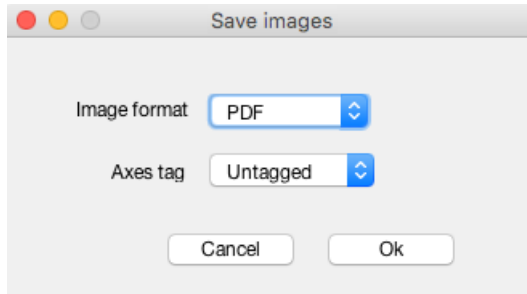
To save a project the user can click the icon , press CTRL + S (cmd + S), or go to Project > Save > Save project. A dialog box will pop-up to indicate a folder where to save it. Projects are saved by default with names starting with prefix 'Aracne\_' followed by the name of the image, although it is possible to modify them as the user desires. Two files are saved, one of them being the program file itself, with extension '.mat', and a '.txt' file which contains information about the images and the processing. Here is stored the information about the image, the processed region, and the parameters used. The results of the thread counting are also included, as well as the statistical values.

#### 4.1.2 Opening an existing project

To load an existing project the user can click in the icon , press CTRL + O (cmd + O), or go to Project > Open project.


## 4.2 Saving results images

The user may be interested in saving the images after some results are obtained, this can be accomplished by pushing the icon  or by going to Project > Save > Save actual images.



**Figure 4.1** Some options that the user can choose for the images to be saved.

A window like Figure 4.1 will show up, in which the user can specify some settings for the images to be saved such as the format and labelled axes, although this last option is not allowed when the results are overlapped with the processed image. Currently, available formats are .pdf, .png, .jpeg, or .tif, and axes can be labelled whether in cm or in inches.

After choosing between these options, the images will be saved in the folder indicated. In Table 4.1 we include a description of the saved images. All images names start with 'Aracne\_' followed by the name of the original image processed, and finishing with the text included in the left column in Table 4.1. If the overlap option () is activated, then the overlapped images will be saved.



**Table 4.1** Description of the saved images.

<b>Image name termination</b>	<b>Image description</b>
_original	Original image.
_selectedProcessedRegion	Selected area of the original image processed.
_HorizontalThreadsAngle	Image with the angle deviation of the horizontal threads. If the overlap option is activated then this image will be renamed as _HorizontalThreadsAngleOverlap.
_HorizontalThreadsCount	Image with the thread counting of the horizontal threads. If the overlap option is activated then this image will be renamed as _HorizontalThreadsCountOverlap.
_HorizontalThreadsCountHist	Image with the histogram of the thread counting for horizontal threads.
_VerticalThreadsAngle	Image with the angle deviation of the vertical threads. If the overlap option is activated then this image will be renamed as _VerticalThreadsAngleOverlap.
_VerticalThreadsCount	Image with the thread counting of the vertical threads. If the overlap option is activated then this image will be renamed as _VerticalThreadsCountOverlap.
_VerticalThreadsCountHist	Image with the histogram of the thread counting for vertical threads.
_Contour	Image with the contour plot of the power spectral density results.
_Mesh	Image with the 3D plot of the PSD results.



## 5 Parameters

---

A correct set of parameters is crucial for a better performance of the counting algorithm. However, to understand their importance one must first know how the algorithm works. We next provide an overview of it including an explanation of the parameters involved at every step.

### 5.1 Algorithm overview

The software starts centering a square window of size  $(m \cdot 1) \times (m \cdot 1)$  cm in the X-ray image where  $m$  = 'Side of the window', and then computes the thread counting and the threads' angle deviation. Then, this window is shifted  $n \cdot m$  cm in both horizontal and vertical directions, where  $n$  = 'Shift fraction' is a fraction of the window size  $m$  to compute the thread counting over the whole image. For each of these shifts, the algorithm computes the number of threads and their deviation within the  $m \times m$  square.

The result of each window will be a color pixel in the output images, being these horizontal and vertical threads counting, and horizontal and vertical angle deviations.

These thread counting and angle deviation estimation algorithm is used in the computation for every  $m \times m$  square. One important parameter of this algorithm is the number of FFT points,  $128 \times p$  where  $p$  = 'Resolution scaling', computed. A low value for this number of points involves less computational demanding operations but provides a worse resolution. On the contrary, a high value for  $p$  provides a good resolution but leads to a larger execution time.

The algorithm returns an estimated value for every thread counting value. Then the maximum values for vertical and horizontal threads are searched and these values are the ones showed in the results and used in the histograms. The algorithm searches for the maximum values within a given range, ['Minimum vertical counting', 'Maximum vertical counting'] and ['Minimum horizontal counting', 'Maximum horizontal counting'] for the vertical and horizontal threads respectively.

The thread counting for vertical threads is performed studying variations along the horizontal axis. However, the fabrics usually exhibit a rotation that slightly varies along

the image. Hence, the algorithm first computes the angle around the horizontal axis that presents the maximum vertical thread counting. This angle is searched within a  $\pm\alpha$  degree range ( $\alpha$  = 'Maximum deviation angle'). Since the whole image may have a rotation, the program allows for a rotation,  $\beta$  = 'Rotation angle', of the whole processed window before looking for these maximum. All these angles are in degree units and are positive counter clockwise.

## 5.2 Possible and Default values

The program include a set of parameters that can be customized by going to Processing parameters > User's parameters (Figure 5.1). The description of these parameters and possible values are discussed in the following.

- **Resolution scaling (p).** The resolution of the thread counting can be observed as the width of the bars in the histograms, depends on this parameter. The obtained resolution can be computed as  $s/(p \cdot 128)$  where  $s$  is the number of pixels per centimeter of the image. The higher the value of  $p$  is, the better the resolution we get, but the required computing time also increases. The default value for the resolution is 4 although its possible values are integers within the range 1 - 16. Powers of two are recommended to optimize the computing time, hence, recommended values are 1, 2, 4, 8 or 16.
- **Side of the window (m).** The size of the square window used as a moving mask to study the threads frequencies is allowed to change in steps of 1 cm, therefore it can be viewed as the number of  $1 \times 1$  cm. Accordingly, parameter  $m$  can be between 1 and 10, and its default value is  $m = 2$ .
- **Shift fraction (n).** In every step of the algorithm execution, the window of size  $m$  is shifted  $n \cdot m$  to compute new values. The default value for this displacement distance is set at  $n = 0.25$ , and it should be in the range  $0.01 \leq n \leq 1$ . Note that the maximum shift is 1, meaning that there is no overlap between successive windows while counting.
- **Minimum and maximum vertical and horizontal counting.** ( $[v_{\min}, v_{\max}, h_{\min}, h_{\max}]$ ). By default, the software performs the thread counting for default values between 4 and 25 threads/cm. Points out of this selected range are regarded as meaningless points, i.e. regions where the counting could not be performed. Note that later, in the histogram the user can further limit the visualization range. But this is just to adjust by zooming the observed range, not changing the frequency counting range. To change the computation range, the user must change these parameters in the program menu and run the thread counting.
- **Maximum deviation angle ( $\alpha$ ).** The counting algorithm looks for the maximum frequency in the horizontal and vertical directions. However, if the image is distorted by a rotation, this maximum frequencies are found in some direction  $\rho$ , where  $\rho$  should be a low value. The algorithm looks for the maximum in any direction

from  $-\alpha$  to  $\alpha$  for the horizontal direction and from  $90^\circ - \alpha$  to  $90^\circ + \alpha$  in the vertical one. The default value for this angle is  $25^\circ$ , and it should be in the range 5 to 45. Notice that this value should not be larger than this value.

- **Rotation angle ( $\beta$ ).** Aracne is programmed to count threads in vertical and horizontal directions, thus, analyzing a canvas with a different distribution of threads would lead to distortions in the counting. However, it is possible to introduce an initial rotation angle if the user previously know that is the case. For example, if all the image is rotated by an angle  $-\beta$ , in the parameters settings it should be indicated to rotate  $\beta$ . Default value is  $\beta = 0$ .
- **Spectral Window.** The user can specify the type of spectral window. Its possible values are 'Hamming', 'Hanning', 'Triangle', 'Rectangle' and 'Blackman-Harris'.
- **Pre-processing type.** Some of the canvases the user analyses might present problems, either by the deteriorated quality of the canvas or by the application of certain materials, thus, a pre-processing algorithm is implemented. A 'STD' pre-processing, which is default, subtracts the local mean to the original image and divides it by its local standard deviation, while a 'Mean' pre-processing only subtracts the local mean. It is also possible to not perform any pre-processing at all if the user wishes so.

**Hint:** The resolution scaling could be set to a low value first to perform some first trials and then, for the final counting estimation, increase its value. Also, a high value for the shift fraction can be set at first and then set it to a lower value for the final counting estimation.

### 5.3 Save and Load Parameters

It is possible to export or/and import a certain configuration of parameters. The current set of parameters can be saved in a .csv file with the name 'Aracne\_Parameters' (although it can be modified) and thus shared between Aracne users, who only would have to import such file. This function was implemented in order to ease the communication between colleague's works.

**Important:** The user should only import .csv files that were previously exported by Aracne, otherwise this action could yield errors.

### 5.4 Preprocessing

Prior to perform the thread counting, the image can be pre-processed. There are three options, no preprocessing, a simple mean correction algorithm and a mean and standard deviation correction method.

Table 5.1 Numerical parameters.

Parameter	Default	Range
Resolution scaling ( $p$ )	4	$p = 1, 2, \dots, 16$
Shift fraction ( $n$ )	0.25	$0.01 \leq n \leq 1$
Side of the window, ( $m$ in cm)	2	$m = 1, 2, \dots, 10$
Maximum deviation angle, ( $\alpha$ , degrees)	5	$\leq \alpha \leq 45$
Rotation angle, ( $\beta$ , degrees)	0	$-45 \leq \alpha \leq 45$
Minimum vertical thread counting, ( $v_{\min}$ in thread/cm)	4	$0 \leq v_{\min} \leq v_{\max}$
Maximum vertical thread counting, ( $v_{\max}$ in thread/cm)	25	$v_{\min} \leq v_{\max} \leq 100$
Minimum horizontal thread counting, ( $h_{\min}$ in thread/cm)	4	$0 \leq h_{\min} \leq h_{\max}$
Maximum horizontal thread counting, ( $h_{\max}$ in thread/cm)	25	$h_{\min} \leq h_{\max} \leq 100$

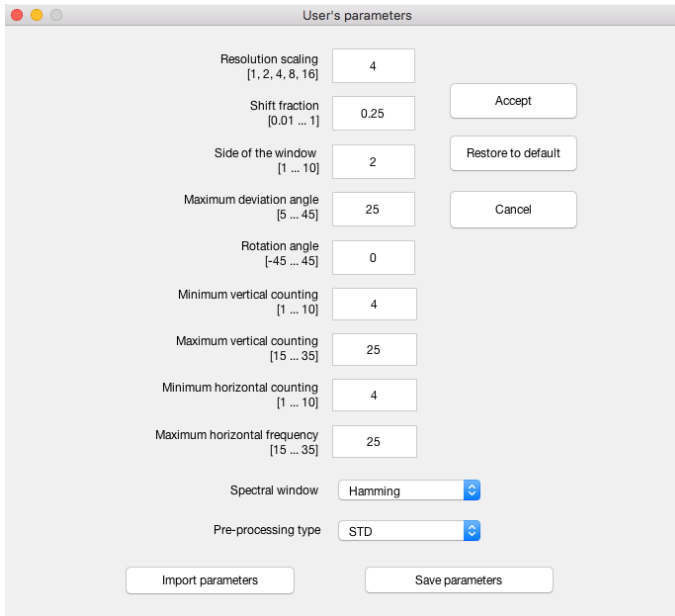


Figure 5.1 Window of the parameters' customization.

# Appendix A

## Image Format

---

In Table A.1 we include some the fields of the format of an example image.

**Table A.1** Fields of the format of an example image.

<b>Field</b>	<b>Value</b>
Filename	[1x88 char]
FileModDate	'05-jun-2014 17:26:28'
FileSize	47857960
Format	'tif'
FormatVersion	[]
Width	7848
Height	6095
BitDepth	8
ColorType	'grayscale'
FormatSignature	[73 73 42 0]
ByteOrder	'little-endian'
NewSubFileType	0
BitsPerSample	8
Compression	'Uncompressed'
PhotometricInterpretation	'BlackIsZero'
StripOffsets	24356
SamplesPerPixel	1
RowsPerStrip	6095
StripByteCounts	47833560
XResolution	508
YResolution	508
ResolutionUnit	'Inch'
Colormap	[]
PlanarConfiguration	'Chunky'
TileWidth	[]
TileLength	[]
TileOffsets	[]
TileByteCounts	[]
Orientation	1
FillOrder	1
GrayResponseUnit	0.0100
MaxSampleValue	255
MinSampleValue	0
Thresholding	1
Offset	8
Software	'Adobe Photoshop CS3 Windows'
DateTime	'2013:08:06 09:29:15'
Artist	[1x43 char]
XMP	[1x15295 char]
ITPC	[11x1 double]
Photoshop	[8642x1 double]
DigitalCamera	[1x1 struct]