EpiGraph: an open-source platform to quantify epithelial organization.

Supplemental Information.

SUPPLEMENTAL EXPERIMENTAL PROCEDURES

EpiGraph Source code.

The project's code is accessible through Github https://github.com/ComplexOrganizationOfLivingMatter/Epigraph. It is open source and available under GPLv3 license.

Pipeline.

The image processing pipeline entails cells recognition, valid cells identification, graph of cell-cell contact creation, polygon distribution analysis and, finally, GDDH, GDDRV and GDDV5 calculation. These stages are properly described on the subsequent paragraphs.

- Recognition of existing cells

EpiGraph uses as input segmented images from natural tissues or simulations. These images must be built in binary format, where one colour should be presented as the background, forming the body of cells, and the other one the cells outline. Each cell gets assigned a label using MorphoLibJ (Legland et al., 2016). This identifier will allow us to have a record of every cell and his location on the image that it will be necessary for the following steps.

- Valid cells identification

Once we have properly labelled the image, we proceeded to fully analyse it. By default, an invalid region is created. This region is initially defined by the boundaries of the image itself. We got all the cells that fall into these borders and set them as no valid cells. In this way, we avoid an incorrect characterization of polygon distribution due to the lack of real neighbours in the

margins of the image. However, you can add a personalized invalid region selecting as many cells as you want.

In addition, we define another two kinds of valid cells: the 4' valid cells and the 5' valid cells. The former represents the cells that do not have a not valid cell within a distance of four cells connexions. The later kind is formed by the cells having all valid cells in a maximum length of five cells connexions. Mo7 and Mo10 require 4' valid cells and Mo17 and Mo29 5' valid cells.

- Neighbourhood creation

An epithelium can also be seen as a tessellation because there's no space between each cell. Thus, inspecting the number of sides of the cells we can measure the number of neighbours that will surround it. We, therefore, extend a mask from each pixel of the cell with a given shape and size, both selected by the user. Then, we capture all the cells distinct from the actual cell that fall into this mask and add them as neighbours of the concrete cell. Afterwards, we create a neighbourhood network, modeling each cell as a node and connecting two cells with an edge, if they are neighbours.

- Polygon distribution analysis

The polygon distribution is defined by the number of sides of every valid cell. We specify three different areas: the global zone formed by all valid cells, the region in which 4' valid cells are contained and the territory in which 5' valid cells are placed. For Mo7 and Mo10, we use a path of length 4 (4' valid cells as orbits nodes, and valid cells as its branching nodes), and for Mo17 and Mo29 the area is defined by the cells contained within the path of length 5 (5' valid cells as orbits nodes, and valid cells as its branching nodes). Finally, we calculate the polygon distribution regarding the cells involved in the graphlet calculation.

- GDDH, GDDRV and GDDV5 calculation

We have adapted Graphlet degree Distribution agreement Distance (GDD) from Yaveroğlu et al. (Yaveroğlu et al., 2014) to fully integrate it with FIJI java environment. We use GDD to compare two images getting how similar or dissimilar they are. In particular, minimum distance value (0) means two images

are equal, and maximum distance value (1) specifies that they are very different in terms of graphlets. We compute all the graphlets in which the valid cells participate. Then, we use only the graphlets in which 4' or 5' valid cells are included (depending on the set used). We defined three references to compare with the real images. The first one is Graphlet degree Distribution agreement Distance Random Voronoi (GDDRV). A random Voronoi (RV) is generated from 500 random points establishing related Voronoi cells. Then, we computed the distance between twenty RV images and the real image getting only the mean of these twenty distances as final reference. We also calculated Graphlet degree Distribution agreement Distance Hexagons (GDDH), which measure the different between a given image and a regular tessellation of hexagons, in a similar way than GDDRV. And, finally, we computed the Graphlet degree Distribution agreement Distance Voronoi 5 (GDDV5) by comparing a given image with twenty diagrams 5 from the CVT path and getting its mean.

A full set of tutorials explaining how to install and use EpiGraph is available at EpiGraph's wiki (https://imagej.net/EpiGraph). In addition, Movie S1 summarizes a general example with default options.

Graphical user interface.

ImageJ's distribution FIJI (Schindelin et al., 2015, 2012) provides a framework which offers a range of functions and a suitable predefined application programming interface (API). Furthermore, it allows users without programming expertise to execute and exploit all their functionalities. EpiGraph is prepared for heavy processes. On these cases, some operations such as "calculate graphlets", perform on background giving the user the possibility of move around the application smoothly. Besides, a progress bar displays the evolution of the task in case it could take longer than expected. FIJI works on the three main Operative Systems (Linux, Mac and Windows) without even require Java installation since it is usually embedded in the program. FIJI describes different ways to be operated by external software. One of them is constructing a plugin, which is used by EpiGraph as basic structure to start building. Using Java

Swing GUI libraries, we have designed the three windows of the EpiGraph plugin.

- Main Window

The program starts after clicking on *Plugins>EpiGraph*. The first window that appears is the Main Window. It contains a table of processed images (therefore, initially it is empty). Graphlets calculated data could be instantiated into a table that is organized by columns. A table's row is composed by these columns: Colour, label, GDDH, GDDRV, GDDV5, percentage of hexagons, radius of shape, type of shape, kind of graphlet calculation and the selection of the graphlets results. *Colour* column lets you pick the colour of the point from the palette to be visualized later; *Label* shows the name of each image data; *GDDH*, *GDDRV* and *GDDV5* represent the GDD of the image against hexagons, random Voronoi diagrams and Voronoi 5 diagrams from CVTn, respectively; *Percentage of hexagons* contains as his graphlet name said, proportion of hexagons for all cells involved in graphlet calculation; *Radius* specify the shape's size used to calculate neighbours; *Shape* characterizes the type of form used to calculate neighbourhood; *Kind of calculation* displays which set of graphlets was used. Finally, *Select* column lets you by means of a check box, the possibility of visualizing or not the calculated data in a 3D viewer, and in the same way, remove the selected rows if you click on the button *remove rows*. Additionally, you do not need to calculate every single time the graphlets properties, you can import your own dataset properly formatted from an Excel file, using *import table* button. Likewise, once you have already analysed several images, you could export your information into an Excel file, using the *export table* button. In addition of exporting the data contained into the table, as tip, you export the polygon distribution of the cells involved in the graphlets calculation. Lastly, you have the option of representing all table's rows, in which visualizing check box was typed as true. For that, you should click on *visualize* button to launch the mentioned 3D viewer.

- Image processing Window

This window will be automatically triggered when a supported file format was correctly selected, after click over the *open* main window button. This window is composed by a button panel with different processing options, a canvas with the loaded image embedded in it and a polygon distribution legend. The only enabled button when the window just opened will be the mode in which you label the image. You can choose labelling this image considering 8 connectivity or 4 connectivity and then, click on *label image* button to execute it. After that, other buttons become enabled. These buttons are classified in 3 panels:

- *Region of interest*. The main objective of this panel was selecting regions of interest to only process valid regions and discard the invalid ones. *create ROI* and *select cells* buttons let you select various rectangular (default) regions and picking cell by cell respectively using the FIJI's ROI manager tool. You could combine repeatedly both options to establish a valid region in which operating. In addition, you might modify the ROI selection shape from rectangular to another one, using FIJI's control panel. On the other hand, we had the option to choose an invalid region by picking over it, after clicking *add invalid regions* button. This action only lets you store a set of invalid ROIs at a time, so if you wanted to add an invalid region after saving previous invalid regions, you will delete the former to include a new one.
- *Neighbourhood*. It lets you the possibility of visualizing valid and no valid cells, invalid regions and polygon distribution values from the current image. To calculate the neighbourhood you should select, using the number picker (located close to *radius*), a size of shape in pixels and a specific geometrical shape using the picker above (located next to *shape*). This selection depends on the wide of the cells outline, you could only choose between circular or square shape, but using any size of pixels. When the parameters have been selected, *test neighbours* button can be pressed to calculate the neighbourhood for the valid cells belonging to the ROI. If the ROI is the whole image, a column next to the legend of polygons shapes, headed by "Graphlets", would be filled with

the polygon distributions of the valid cells. Otherwise, if the ROI is a subsection of the image, the cited column would be filled with the polygon distribution of the all valid cells participating in the ROI graphlets calculation. In addition, a new column appears next to “Graphlets” headed by “ROIs” that would be filled by the polygon distribution of the valid cells belonging to this ROI.

In general, the legend will be completed with the proportion of polygons and an overlay would be displayed over the canvas, representing the type of polygons with a specific colour.

In addition, *toggle overlay* button lets you visualize or not the overlay created by *test neighbours*. Black colour labels the invalid regions and the cells outlines at the overlay, dark grey marks no valid cells, and the rest of colours (represented in the polygon distribution legend) are reserved for valid cells. It is important to highlight that the intense colours of the legend displayed over the canvas represent the number of sides of the valid cells into the ROI. The same pale colours represent the polygon distribution of valid cells located out of the ROI affecting to the graphlet calculation.

Graphlets. This panel is designed with the aim of saving graphlets data internally and externally. There is a write section where you can add the data label (image name by default), located after *image label* text. As well, you can select a colour label for our data, using *pick a colour* button. Finally, you choose the appropriated method to calculate graphlets data. The options are: ’26 graphlets (Mo29)’ (default), ‘17 graphlets (Mo17)’, ‘9 graphlets (Mo10)’ and ‘7 graphlets (Mo7)’. As soon as the method has been selected, you can continue clicking *calculate graphlets data* button to acquire data of calculated graphlets. Once this process has successfully finished, data are automatically added on the main window table. Furthermore, clicking over *Export Graphlet data* you can export a ZIP file containing: a JPG image representing the neighbourhood, another JPG image capturing the label of the cells, a CSV file storing all

calculated graphlets data and a .sif that represent the neighbourhood network.

This window has a progress bar that estimates the process state. As a tip, you can modify the image zoom by pressing control and rotating the mouse wheel at same time.

- Visualizing Window

We count with a 3d viewer to display our calculated results stored in the main window's table. When a row´s checkbox is ticked, it will be plotted. To develop this window, we have used an open source library named Jzy3D (http://www.jzy3d.org/) that is able to create different graphical representations. In particular, we make use of the 3d scatter plot class.

This window will be displayed after click over *visualize* button (located in the main window), even if your table is still empty. This window is composed by the scatter plot figure located at left and a set of components to modify the appearance of this figure at rigth. The plotted figure is delimited by a 3D box with 3 axes: Percentage of hexagons, GDDH and GDDRV by default. These 3 axes can be replaced by any of the following configurations (X-Y-Z axes) using the drop-down list with the label *Axes of figure* (upper right corner of the window):

1- GDDH-GDDRV-Percentage of hexagons (default)
2- GDDH-GDDRV-GDDV5
3- GDDH-GDDV5-Percentage of hexagons
4- GDDRV-GDDV5-Percentage of hexagons

The Percentage of hexagons axis encapsulate values between 0 – 100, and the others between 0 – 1. These limits can be modified for zooming the individual axes. We used 3 *rangeSliders* to choose the range for each axis to be represented due to the limitations of the Jzy3D library. This library only provides you the possibility of zooming the Z axis turning the mouse wheel. You can visualize the three *rangeSliders* (one per axis) just below the *Axes of figure* drop-down list.

Into the scatter plot you can observe by default our CVTn path, displayed as individual dots: The darkest dot, represents the average of first iterations in CVTn and the lightest one the average of last iterations of CVTn. You have the option to disable the visualization of these references clicking on the checkbox 'Show reference' (just below of *rangeSliders*), to only display calculated data. Besides, it is available the adjustment of the size of the dots modifying the position for slider bar, just located down the mentioned checkbox. Distinguishing the different modes in which graphlets data of CVTn reference can be calculated, a drop-down list can be deploy clicking over the label *Motifs of CVTn reference*. This list, let you the possibility of representing CVTn path depending on the method in with graphlets are calculated: Mo29, Mo17 (by default), Mo10 and Mo7. We also have defined four possible combinations of graphlets that can be computed by EpiGraph: 26 graphlets representing 29 cellular motifs (Mo29), 17 graphlets that represent 17 cellular motifs (Mo17), 9 graphlets on behalf of 10 cellular motifs (Mo10) and 7 cellular motifs represent by 7 graphlets (Mo7). Mo7 and Mo10 are formed by graphlets of maximum 4 nodes, while Mo17 and Mo29 make use of graphlets of maximum 5 nodes. Mo7 and Mo10 are very useful when images have a small number of cells, since both require a less number of cells than Mo29 and Mo17.

The dots showed in figure from the Main Window can be modified changing the colour box in the Main Window's table. In addition, a .png figure screenshot can be saved, pressing *Export view* button. For more detail, different angles and modes of visualization are available. For instance: clicking the figure and moving the mouse will turn the vision angle, and double clicking on the figure will automatically perform a 360 degrees' rotation.

Functionalities.

Along the program execution process, we will have the capability of developing a set of functionalities:

- Label image

Once you have selected an image to process, this image is binarized and the background is analysed to consider what is the cells colour and outlines colour. If the number of white pixels is higher than black pixels, white pixels will be

considered as cell's body, and vice versa. After that, each cell's body region of the image is assigned with a unique label. Thanks to the extensible architecture of FIJI, in which you can install plugins to add additional functionalities, this process of labelling regions is made using MorphoLibJ (Legland et al., 2016) funcionalities. Specifically, using the connected component labelling that transforms a binary image into a labelled one by assigning a specific number to each connected component.

- Testing neighbours

As mention on previous section, EpiGraph is able to calculate the polygon distribution with a given pixel radius and an element's shape. To ensure that the neighbourhood is correctly captured, we empower the users to verify if the image has the right parameters by themselves. The polygon distribution of the image will appear with numbers at the left side of the window and, also, painting each cell with a colour representing its number of sides. As well, dark grey almost black will represent no valid cells. Depending on the number of graphlets you'd like to see, the affected zone will be established by a group of cells or another. These internal calculations for neighbourhood will be exploited by *calculate graphlets* module, while parameters are not modified since this step.

- Create ROI

Though you may analyse the image as it comes, you could also process only a region of interest (ROI). Through FIJI's Roi Manager we can manage the ROIs, saving and doing operations between each other. Nevertheless, we have only selected as default operations within EpiGraph: rectangle (or any available shape) and multipoint selection. The former creates a rectangle selection (by default) and adds all the cells that fall inside it as valid cells. The latter, enables the multipoint function and it lets select individual cells. Yet, you might want to change these default form of selection, going straight to FIJI's main window and pick any of the toolbar.

-Pick invalid regions

Owing to your image may present artefacts as false valid cells, we provide the possibility to mark these zones as invalid region. All valid cells surrounding the mentioned invalid region will be considered as "no valid cells". On this behalf, it

is possible select, with the multipoint tool, several areas of the image to make them over into invalid sections. To validate this action, picked invalid region should have the same colour than cells' background.

- Calculate graphlets

The main course of EpiGraph is the graphlet comparison. It is supposed to be the final step of the algorithm and pipeline. It begins checking if there are any selected cells or ROIs. Then, in case any configuration has changed, we re-compute the neighbourhood, otherwise we take the information from a previous computation. From this neighbourhood and valid cells, it calculates the graphlets for the involved network of neighbours. As mentioned on previous sections, it would be incoherent not to filter the graphlets, so we refine it by adding to the final graphlets only the nodes with a fixed distance to the border nodes. We first calculate the total set of graphlets for valid cells, and then, we select a filtering that depends on the chosen type of Graphlets to be implicated: Mo29 (26 graphlets), Mo17 (17 graphlets), Mo10 (9 graphlets) and Mo7 (7 graphlets). The involved orbits nodes for graphlets calculations will be referred as 4' and 5' valid cells. When we have the final graphlets, we calculate the three distances (GDDH, GDDRV and GDDV5). Depending on the number of graphlets selected, a variable number of orbits are used on the comparison. At last, the results will be added automatically to the table on Main Window.

- Visualization

Visualizing results properly is a major feature and an important challenge to interpret results. Thus, we have embedded Jzy3d chart in a Java's *JDialog* where you can visualize the calculated results three coordinates (any combination of GDDH, GDDRV, GDDV5 and percentage of hexagons) as a point in a scatter plot. Once the points are represented you can compare them with our CVTn reference. On this behalf, you may want to change the illustrated CVTn selecting the number of graphlets used on the computations, although it is advisable to compare images with the same configuration (Mo7, Mo10, Mo17 or Mo29). Additionally, there is a possibility of increasing or decreasing the size

of the dots and zoom manually each axis. Finally, you can export the actual view of the chart to an image file.

Dependencies.

FIJI is designed to add functionalities via several ways, one of them is Plugin. In order to have a simple control of dependencies and project settings, it uses Maven. Maven is a tool that is broadly used for supervising and build Java-based projects. It integrates several tools, such as Javadoc, to make easy all the programming steps. Likewise, dependencies are downloaded and updated when available. Furthermore, it helps you create the package with all your code and zip it in a .jar file that will be the Plugin format of FIJI. All this is achieved through its project object model (POM), which you can shape to your project adding mailing list, issue tracking and more.

Supported file formats.

EpiGraph's input is an image. This image should be properly segmented and grey-scale, still, there are several options within EpiGraph to configure it depending on the type of the image. We allow images with different size of borders, since it can be permitted to increase the radius of the mask, changing the way in which neighbourhood is built. We make no distinction between colour depth, supporting images from 8-bit to 32-bit. However, it is mandatory that images have to be binary images, where one colour is presented as the background, forming the body of cells, and the other one the cells outline. Regarding the image file extensions, we entrust FIJI the image opening and the supported files as well, so individual image file extensions allowed by FIJI would be supported in EpiGraph. We cannot admit any sequence of images and single images with high resolution due to problems with complexity in EpiGraph processing tasks. Accordingly, we limit the maximum of 3000px to either height or width of the image, for a correct execution of the program.

Quick step-by-step EpiGraph's usage.

- Installation

Through FIJI/ImageJ update site:

The usual way to install a FIJI plugin is through his on-site updater. It's usually located on *"Help > Update FIJI"*. Once, it's open, you click on "*Manage updates sites*", look for EpiGraph and tick the checkbox next to it. Finally, "*Apply changes*" and you should have successfully installed this plugin. With this option, you can automatically get the latest version of EpiGraph.

Manually:

On the other hand, you may just want to download the .jar file from http://tinyurl.com/EpiGraph-PlugIn (or even generate your own .jar from source code) and install it manually. On this purpose, you could click either *Plugins>Install or Plugins>Install PlugIn*. So, you select the provided .jar and you should be able to run EpiGraph.

- Simple example of a complete analysis

Calculate GDDH, GDDRV, GDDV5 and percentage of hexagons from a given image and visualize it in the 3D visualizer.

1. Select *open*.
    a. Select a supported image.
    b. If the image is correct, a window with the image will appear.
2. Pick connectivity:
    a. Select the connectivity of your image. Usually 8-connectivity.
    b. Press *label.*
3. Default configuration:
    a. Radius of 3 pixels.
    b. Circle shape.
4. Calculate graphlets:
    a. Write a name for your image.
    b. Pick a colour for your image.
    c. Select in the combo box to *Mo29 (26 graphlets).*
    d. Press *calculate graphlets*. This will calculate GDDH, GDDRV and GDDV5.

e. Once it is finished, if name box is not empty, data would be added to the main window's table.

5. Export graphlets data:
    a. EpiGraph create a zip with several files. Obtains an image with all the cells labelled with their corresponding identifier. Another image representing the polygon distribution and, finally, a '.csv' file with the graphlets for all the cells.
    b. Press the *Export graphlets data* button.
6. Visualize your results:
    a. Return to the main window.
    b. Click on *visualize.*
7. Export view:
    a. Staying at the visualizing window, you always have the option of exporting the actual view into a '.png' file.
    b. Press *export view*.

Congratulations! You have done the complete pipeline. Now, check where is your image regarding the CVTn scale. You should analyse whether or not your image is near the CVTn (reference).

<u>- Create region of interest</u>

In the case you would like to test a given region of your images, you can create a region of interest (ROI). We already have an image opened and labelled within EpiGraph.

1. On the image processing window:
    a. Press *Create ROI*.
    b. Although, the default ROI is square shaped, you may want to test another shape. You can now return to FIJI application and select any of the existing ones on the toolbar.
    c. Click *done*. Your ROI is managed and stored in FIJI Roi Manager.

You have already created your ROI, nonetheless, you cannot see how affect it to the image yet.

2. Press *Testing neighbours:*

a. An image representing each cell by its number of neighbours will appear as overlay in your image. Remember that the valid cells into the ROI, appear with intense colour, and the pale colours represent the rest of valid cells that affect the computation of graphlets. These cells will also be represented in the polygon distribution legend.

Once this procedure has been accomplished, you could continue the execution in the same way than before: graphlet calculation, exportation of results and visualization.

- Combine ROIs

Since regions of interest are managed by the ROI manager you could add several ROIs to the image. You have already opened the image and properly labelled.

1. Press *Create Roi*:
    a. Select a left region. You should not select any of the cells on the right yet.
    b. *Done*.
2. Select individual cells:
    a. Click *Select cells*.
    b. Pick only cells on the right.
3. Test your setup:
    a. You should be able to see the cells inside that fall into both ROIs.
    b. Press *test neighbours*.
4. Oh! You realize you have made a mistake:
    a. You do not want the first ROI.
    b. Go to the ROI manager.
    c. Select the first ROI and click Delete.
5. Test the new selection:
    a. You now will represent only the existing ROI on the right.
    b. Click *test neighbours* again.

If you create some ROIs, the select cells will be all the cells that fall into any of the ROIs. The logical operation would be an OR, so ROIs could be disjoint.

- Selection of the number of motifs

There are 4 possible configurations, now we will be creating an analysis with Mo17 using only 17 graphlets.

1. Open an image with only 4-connected elements and borders with 3 or more pixels (for a different example).
2. Label it:
    a. Select now 4-connectivity.
    b. Click *label*.
3. Test if neighbours are correct:
    a. Since you do not know what the radius pixels you should select, test various shapes and radius in order to see if the cells with 6 neighbours are really 6-sided cells.
    b. Change parameters and click on *test neighbours* to make sure the neighbourhood is fine.
4. Mo17:
    a. With this configuration, we have seen that there are no strange 'jumps' between iterations of CVTn, so you will be able to see a trustable result.
    b. Hold in the combo box the default option, *Mo17 (17 graphlets)*.
5. Calculate graphlets:
    a. Add a proper name to your image.
    b. Press calculate graphlets to add results to the table.
6. Visualizing:
    a. Change the colour of your row to a pink-ish colour. This will let you differentiate your data from the reference, which is black, grey and almost white.
    b. Click on *Visualize*.
    c. At default, your reference is *Mo17 (17 graphlets).* So, preserve this option in the combo box.
    d. Modify the axes of figure to 'GDDH-GDDRV-GDDV5'.

e. Change the range sliders at your right side to get an adjusted visualization per axis.

You will see that the reference has changed and your image is probably alienated to the CVTn reference. This is because your image is within the CVTn path, otherwise the image organization would be different to our model of reference.

- Import/export table

You may want to save your results somehow, to achieve this, you might export your results to an excel file. As simple as pressing the button export in the main window.

On the other hand, if you would like to continue your session right where you left, you could import your exported excel file to the table and carry on with your analysis. You can add as many .xls files to the table as you want, we allow duplicates. If you want to remove them, just tick them at the *Select all* column and click on *Delete rows*.

Troubleshooting

If the visualization is not working properly, you may need to update FIJI after the installation.

If you find something weird with any functionality of the application, you can send a e-mail to pvicente1@us.es or pgomez-ibis@us.es. Reporting bugs can also be made through Github https://github.com/ComplexOrganizationOfLivingMatter/Epigraph/issues. Known issues are held at the same page.

License Information.

To encourage the sharing of resources, EpiGraph is published under an open-source (GPLv3) license, which can be downloaded from https://github.com/ComplexOrganizationOfLivingMatter/Epigraph/blob/master/LICENSE.

SUPPLEMENTAL FIGURES

Figure S1. Graphlets and orbits configuration. Illustration of graphlets networks used in (Pržulj, 2007). Each graphlet configuration is labelled with G_n, in which, 'n' is the graphlets number (from G_0 to G_{29}). These labels match with the cellular motifs in (**Fig. 1**). Each vertex represents a cell and each edge the connection between two cells. Therefore, the graphs represent the connectivity network for each matched cellular motif. The three graphlets labelled in red (G_{20}, G_{22} and G_{25}) are the ones discarded in our work. The digits (from 0 to 72) into some mauve vertices indicate the orbits number counted for the GDD calculation.

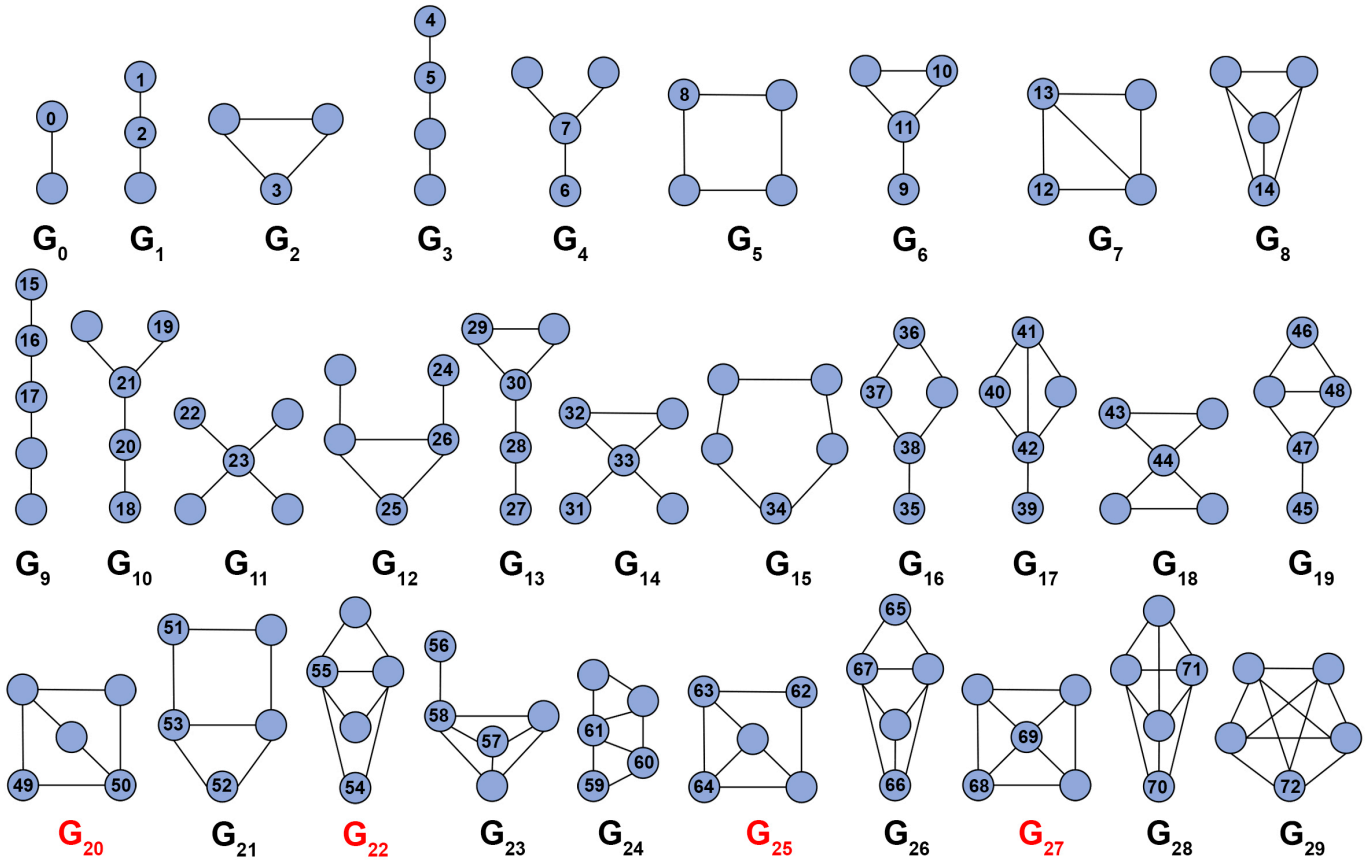

0
G0
1
2
G1
3
G2
4
5
G3
7
6
G4
8
G5
10
11
9
G6
13
12
G7
14
G8
15
16
17
G9
19
21
20
18
G10
22
23
G11
24
26
25
G12
29
30
28
27
G13
32
33
31
G14
G15
36
37
38
35
G16
41
40
42
39
G17
43
44
G18
46
48
47
45
G19
49
50
G20
51
53
52
G21
55
54
G22
56
58
57
G23
61
60
59
G24
63
62
64
G25
65
67
66
G26
69
68
G27
71
70
G28
72
G29

Figure S2. Pipeline for the Graphlet degree Distribution agreement Distance (GDD) calculation. Scheme representing the protocol sequence to calculate the GDD between two segmented images: Eye and Diagram 1. First, a network of cells-to-cells contacts is computed defining the centroids of valid cells as nodes and its connections with neighbouring cells as edges. The nodes are represented with the same colour code as in **Fig. 2**. Second, the graphlets are extracted from the network. This enables the calculation of an index of the distribution of every graphlet. Finally, a comparison between the two Graphlet degree Distributions is performed to obtain the GDD value.

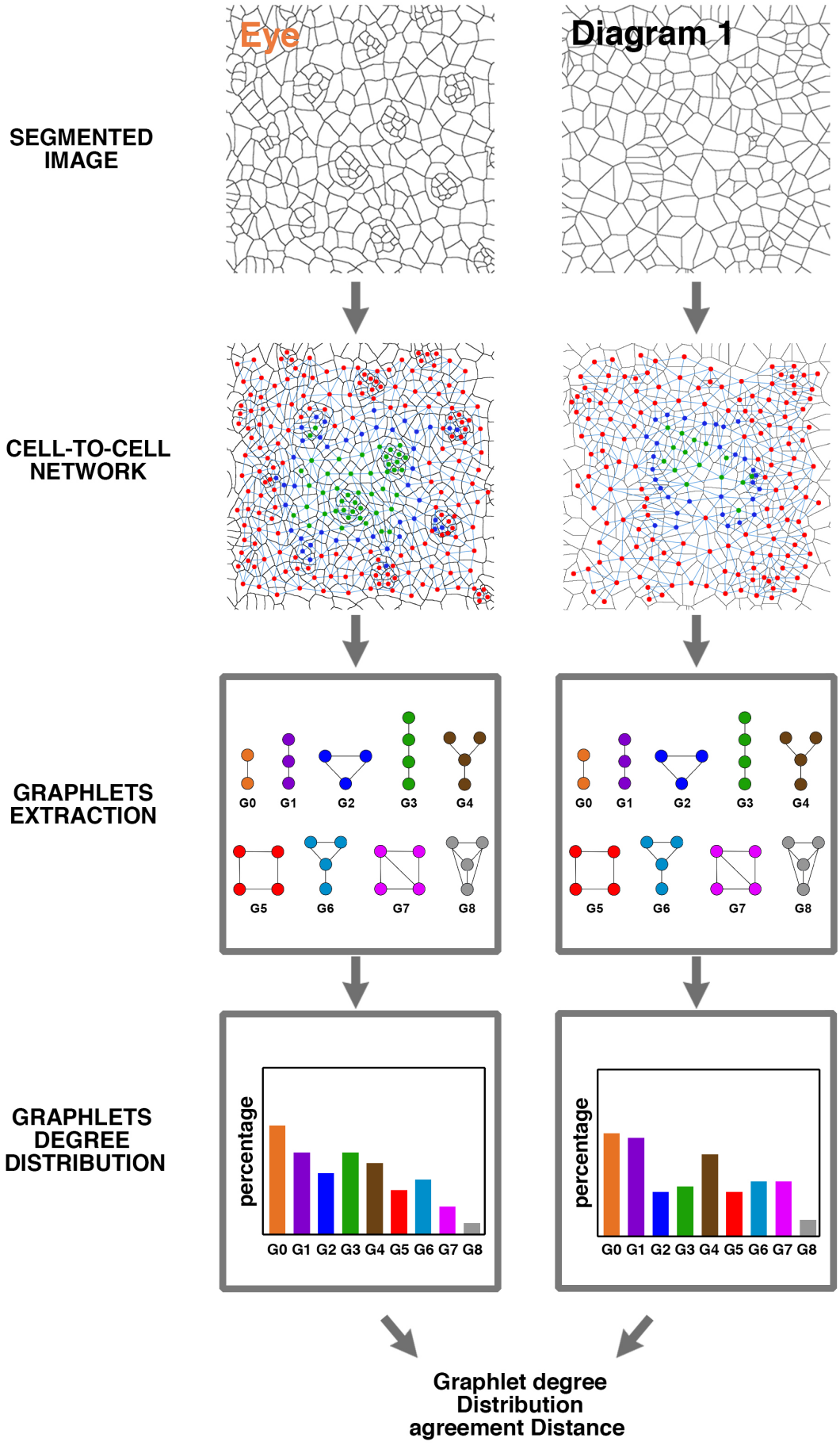

SEGMENTED IMAGE
CELL-TO-CELL NETWORK
GRAPHLETS EXTRACTION
GRAPHLETS DEGREE DISTRIBUTION
Eye
Diagram 1
G0
G1
G2
G3
G4
G5
G6
G7
G8
percentage
G0 G1 G2 G3 G4 G5 G6 G7 G8
Graphlet degree Distribution agreement Distance

Figure S3. Epithelial organization of biological tissues with respect CVT. A-D) Same configurations as (**Fig. 2D-G**) for Mo17 of GDDH, GDDRV, GDDV5 and percentage hexagons, but using CVT path instead of CVTn. The diagrams of the CVT started with iteration 1 and finished with iteration 700. CVT is represented as a grayscale gradient from black, reducing its darkness with the increase of the iterations.

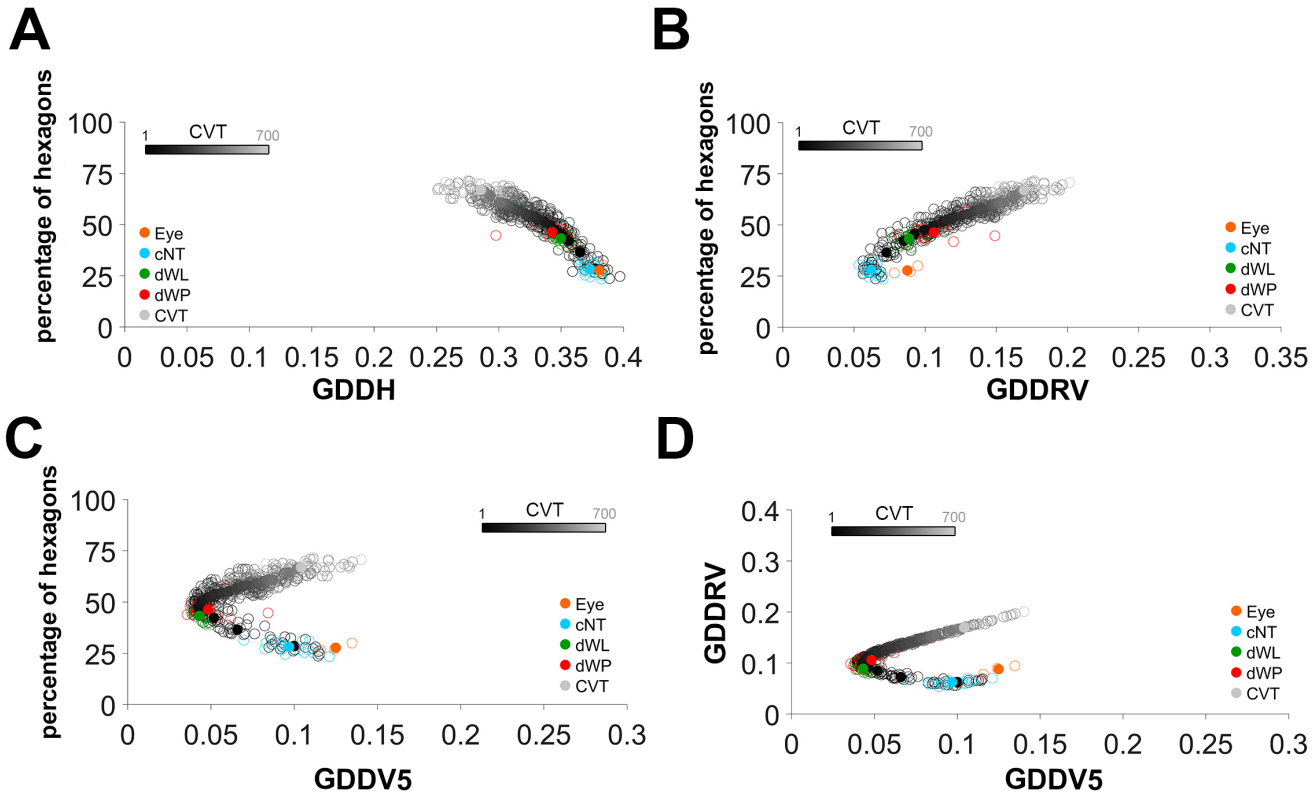
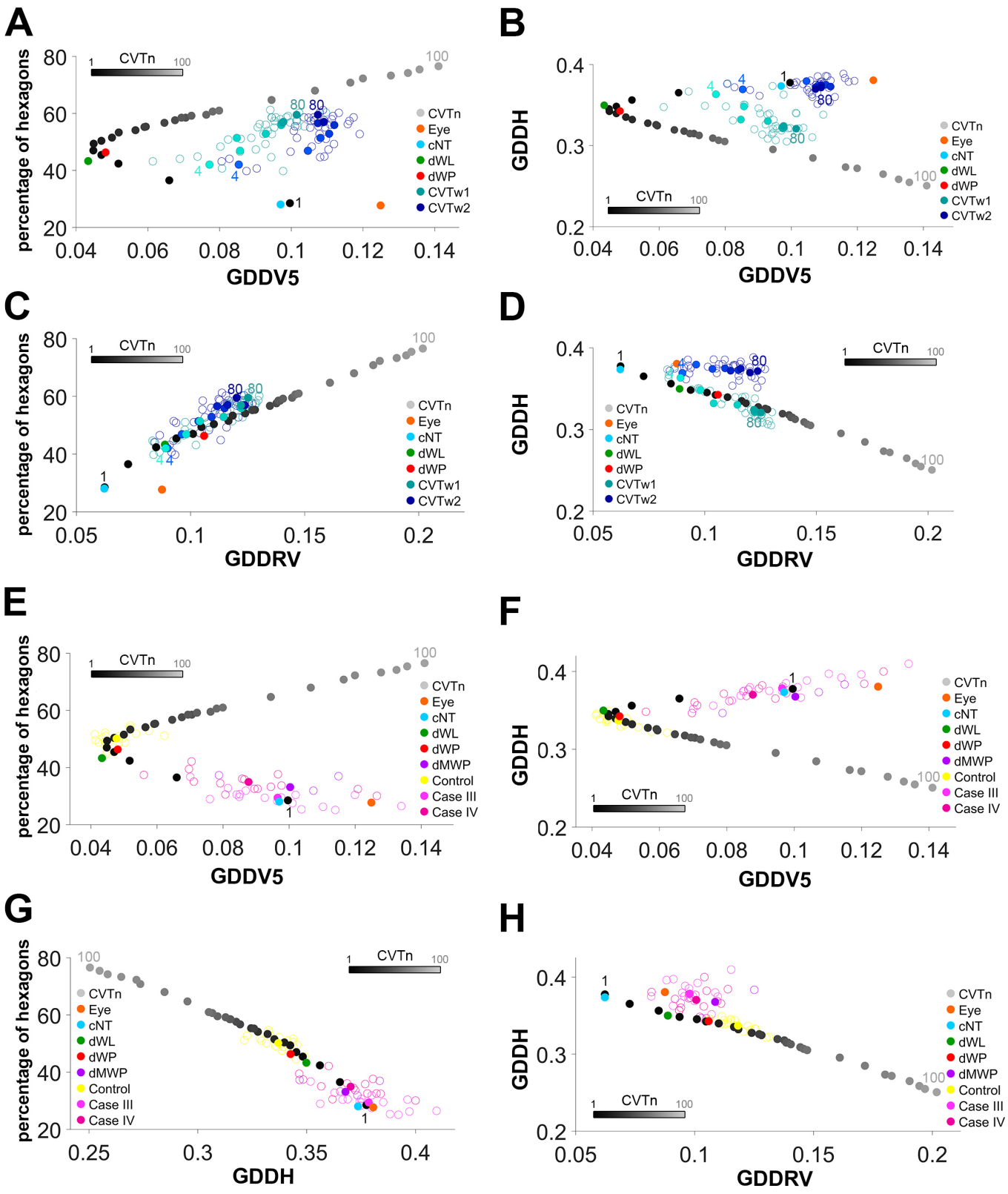

A
percentage of hexagons
100
75
50
25
0
1
CVT
700
Eye
cNT
dWL
dWP
CVT
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
GDDH
B
percentage of hexagons
100
75
50
25
0
1
CVT
700
Eye
cNT
dWL
dWP
CVT
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
GDDRV
C
percentage of hexagons
100
75
50
25
0
1
CVT
700
Eye
cNT
dWL
dWP
CVT
0
0.05
0.1
0.15
0.2
0.25
0.3
GDDV5
D
GDDRV
0.4
0.3
0.2
0.1
0
1
CVT
700
Eye
cNT
dWL
dWP
CVT
0
0.05
0.1
0.15
0.2
0.25
0.3
GDDV5

Figure S4. Extra combinations for Mo17: CVTn, mutants and simulations. A-D) Several GDD combinations (A: Percentage of hexagons vs GDDV5; B: GDDH vs GDDV5; C: Percentage of hexagons vs GDDRV; D: GDDH vs GDDRV) using Mo17 for CVTn, Eye, cNT, dWL, dWP, CVTw1 and CVTw2, represented exactly as (**Fig. 3 B-C**). **E-H)** Values of the percentage of hexagons vs GDDV5 (Mo17) (E), GDDH vs GDDV5 (F), percentage of hexagons vs GDDH (G) and GDDH vs GDDRV (H) for CVTn, Eye, cNT, dWL, dWP, Control, Case III and Case IV. These graph complement (**Fig. 3 H-I**).


A
B
C
D
E
F
G
H
percentage of hexagons
GDDH
GDDV5
GDDRV
CVTn
Eye
cNT
dWL
dWP
CVTw1
CVTw2
dMWP
Control
Case III
Case IV

LEGENDS OF SUPPLEMENTAL FILES

Movie S1. A general example of the usage of EpiGraph. The movie shows how to add the data of a new image using default options.

Table S1. GDD values between pair of images from natural images and Voronoi Diagrams. Graphlet degree Distribution agreement Distance mean between each row and column is shown. Data is divided by the used cellular motifs (Mo17 and Mo29). The used samples are Voronoi 1, 4 and 5: 20 replicates; dWP: 16 replicates; dWL: 15 replicates; Eye: 3 replicates.

Table S2. GDDs and percentage of hexagons of CVTn reference using all the cellular motifs sets. Data are distributed depending on the cellular motifs used (Mo7, Mo10, Mo17, Mo29). Mean and standard deviation of percentage of hexagons, GDDH, GDDRV and GDDV5 are shown, along with their associated diagram. 20 replicates of each diagram are represented.

REFERENCES

Legland, D., Arganda-Carreras, I., Andrey, P., 2016. MorphoLibJ: integrated library and plugins for mathematical morphology with ImageJ. Bioinformatics 32, btw413. doi:10.1093/bioinformatics/btw413

Pržulj, N., 2007. Biological network comparison using graphlet degree distribution. Bioinformatics 23, 177–183. doi:10.1093/bioinformatics/btl301

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., Tinevez, J.-Y., White, D.J., Hartenstein, V., Eliceiri, K., Tomancak, P., Cardona, A., 2012. Fiji: an open-source platform for biological-image analysis. Nat. Methods 9, 676–682. doi:10.1038/nmeth.2019

Schindelin, J., Rueden, C.T., Hiner, M.C., Eliceiri, K.W., 2015. The ImageJ ecosystem: An open platform for biomedical image analysis. Mol. Reprod. Dev. 82, 518–529. doi:10.1002/mrd.22489

Yaveroğlu, Ö.N., Malod-Dognin, N., Davis, D., Levnajic, Z., Janjic, V., Karapandza, R., Stojmirovic, A., Pržulj, N., 2014. Revealing the Hidden Language of Complex Networks. Sci. Rep. 4, 1–9. doi:10.1038/srep04547