Flexible and Reproducible figures using **Inkscape & ImageJ** [& R, Processing…]

Jérôme Mutterer, CNRS

Martin Owens, Inkscape Team

https://gitlab.com/doctormo/inkscape-imagej-panel

2021-04-06

Goal: reproducible figures

- **Inkscape** is a professional grade Free and open source vector graphics editor for GNU/Linux, Windows and MacOS X.
- It can be extended using Inkscape extensions mechanism.
- We present a set of Inkscape extensions that can communicate with widely used other open source software and provide a way of specifying figure content using reproducible code.
- Current included examples are image panels generated using:
  - ImageJ macro
  - R script
  - Processing sketch
- The code for these extensions is public and we invite others to use it, improve it, or extend it (under the terms the GPLv3 license) to generate further interfaces to useful third-party software.

Help us make Inkscape awesome!
English
Log in
Register
INKSCAPE
Draw Freely.
Search website
ABOUT
DOWNLOAD
NEWS
COMMUNITY
LEARN
CONTRIBUTE
DEVELOP
SUPPORT US
INKSCAPE 1.0
Draw Freely.
Island of Creativity by Bayu Rizaldhan Rayes
Download Now!
Get the professional vector graphics editor!
Explore Features
Find out what Inkscape is capable of
Community Gallery
Showcase of creations from the community
Learning Resources
HowTos, Videos, Tutorials and more...

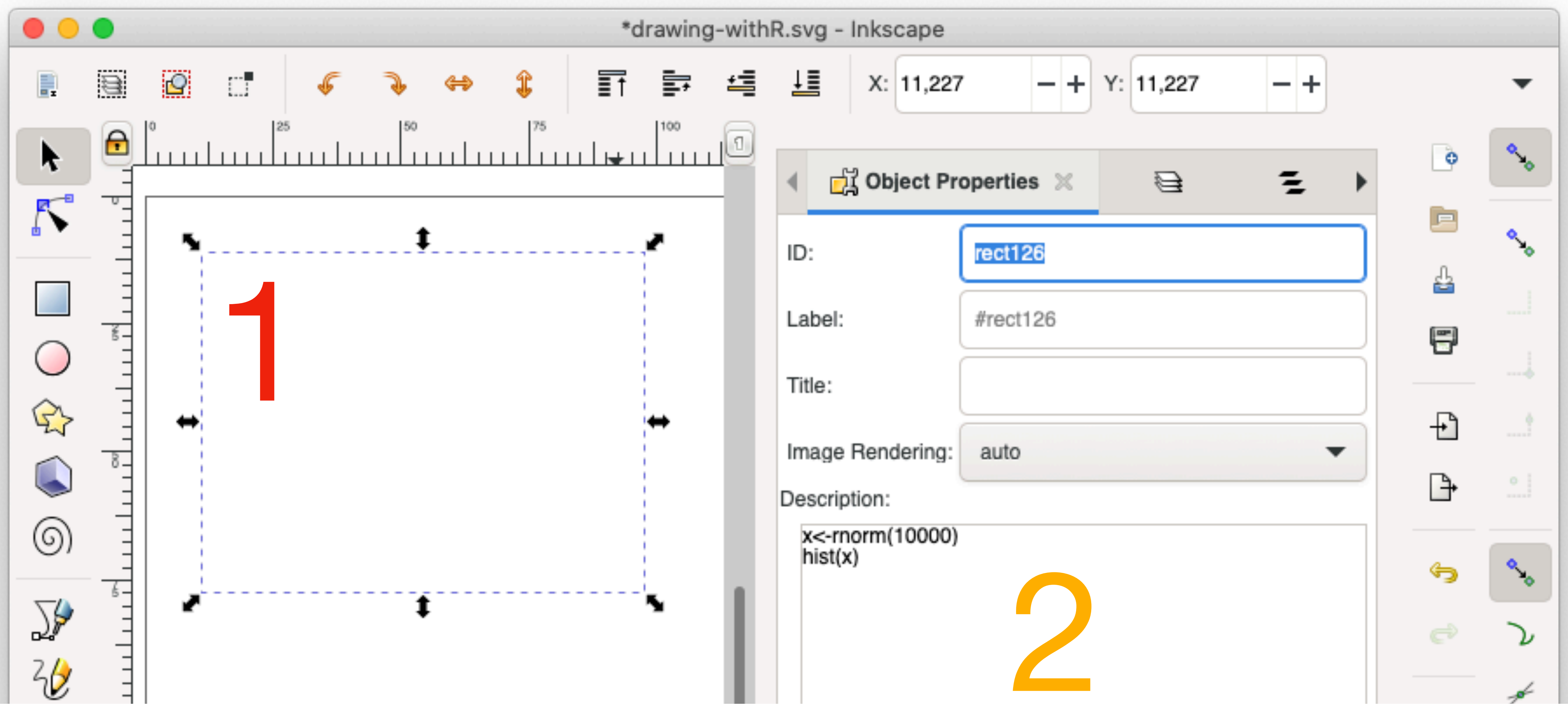
Get Inkscape: https://inkscape.org/

⚠ Requires Inkscape 1.1 or later

Jérôme Mutterer & Martin Owens

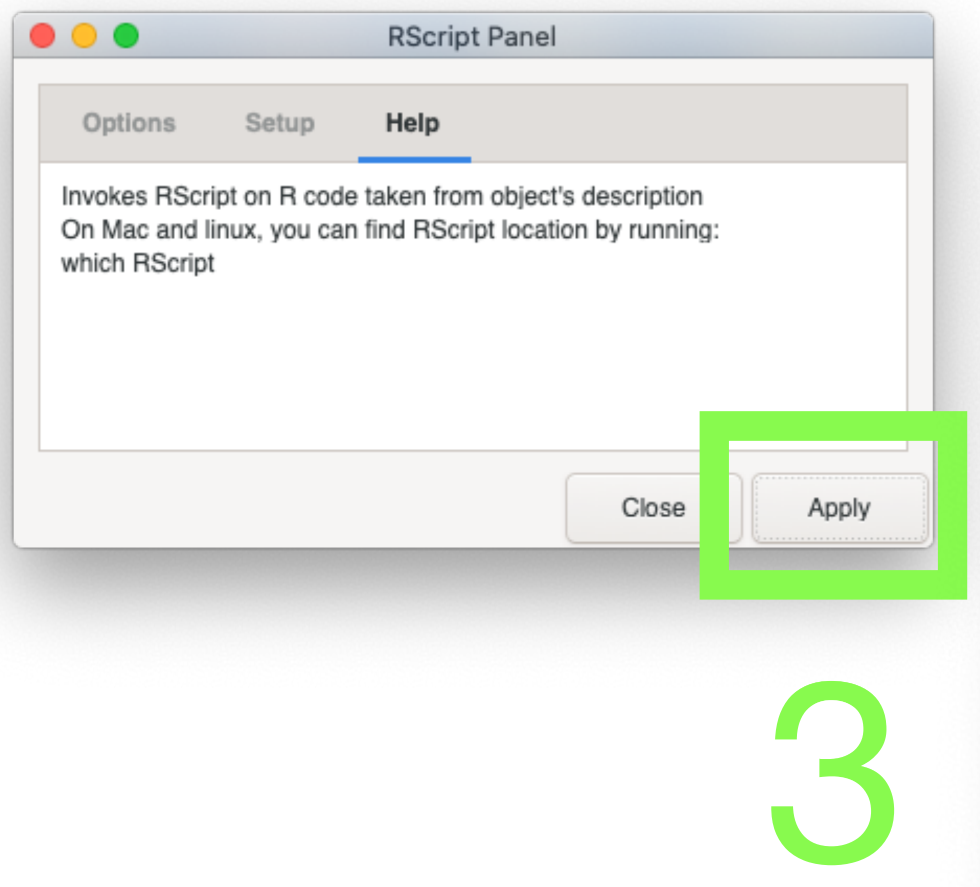
Workflow

1. Specify figure panel by drawing a rectangle
2. Add panel generating code to object's "Description" field
3. Invoke third party software using extension "Apply" button

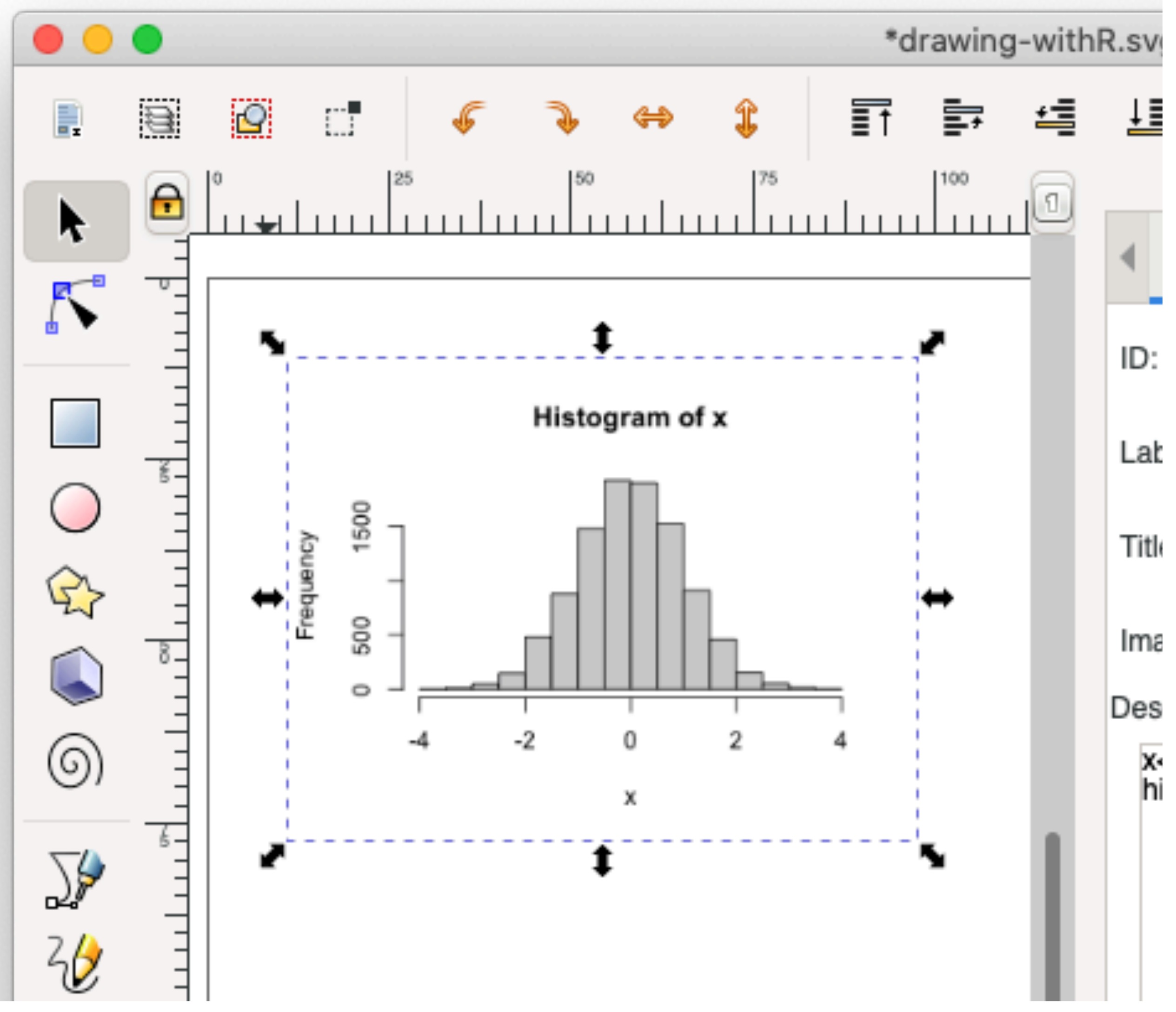


ID: rect126
Label: #rect126
Title:
Image Rendering: auto
Description:

```
x<-rnorm(10000)
hist(x)
```

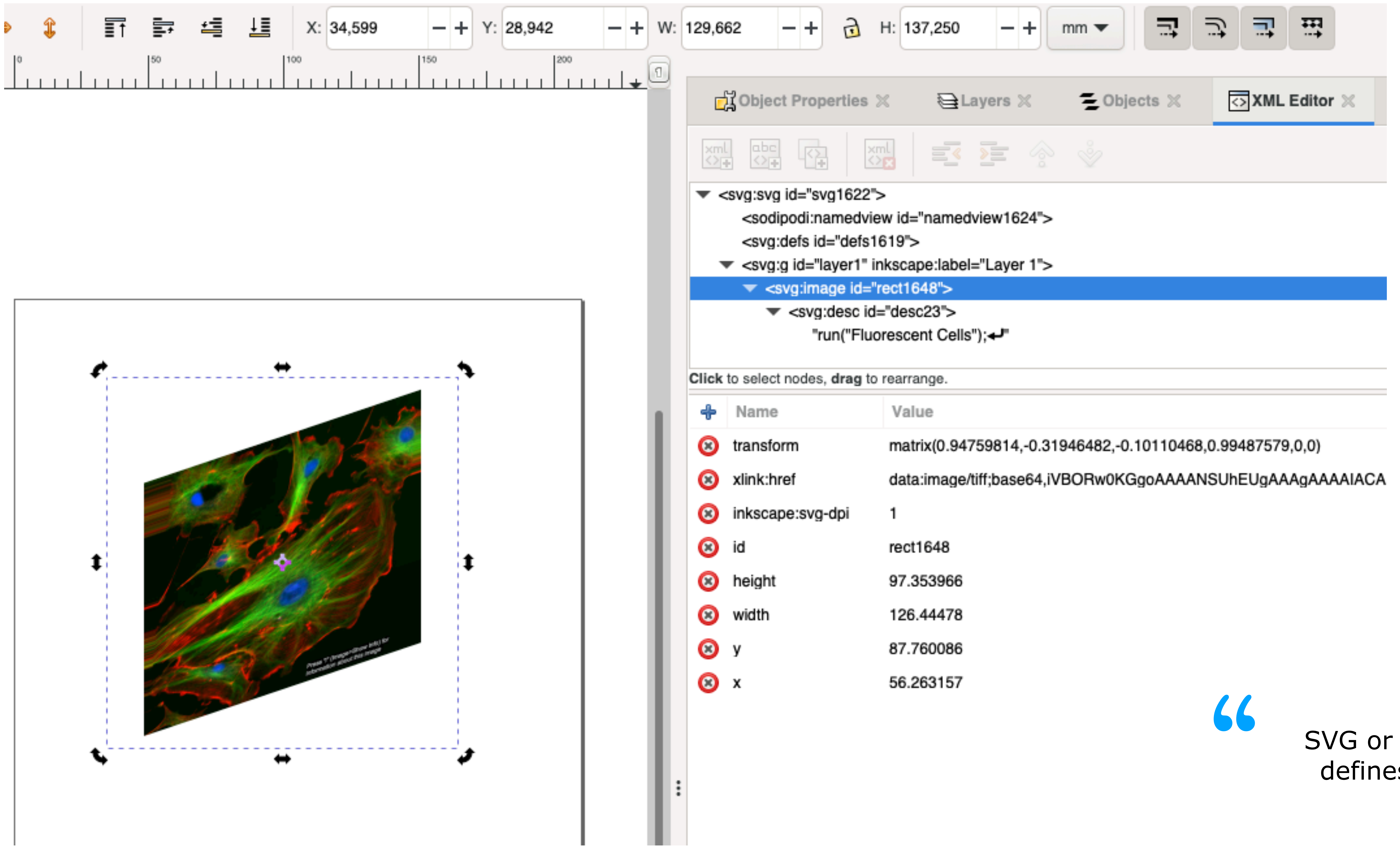


Invokes RScript on R code taken from object's description
On Mac and linux, you can find RScript location by running:
which RScript



Jérôme Mutterer & Martin Owens

Data in svg format



```
<svg:svg id="svg1622">
    <sodipodi:namedview id="namedview1624">
    <svg:defs id="defs1619">
  <svg:g id="layer1" inkscape:label="Layer 1">
    <svg:image id="rect1648">
      <svg:desc id="desc23">
        "run("Fluorescent Cells");↵"
```

Click to select nodes, **drag** to rearrange.

Name	Value
transform	matrix(0.94759814,-0.31946482,-0.10110468,0.99487579,0,0)
xlink:href	data:image/tiff;base64,iVBORw0KGgoAAAANSUhEUgAAAgAAAAIACA
inkscape:svg-dpi	1
id	rect1648
height	97.353966
width	126.44478
y	87.760086
x	56.263157

svg format with:

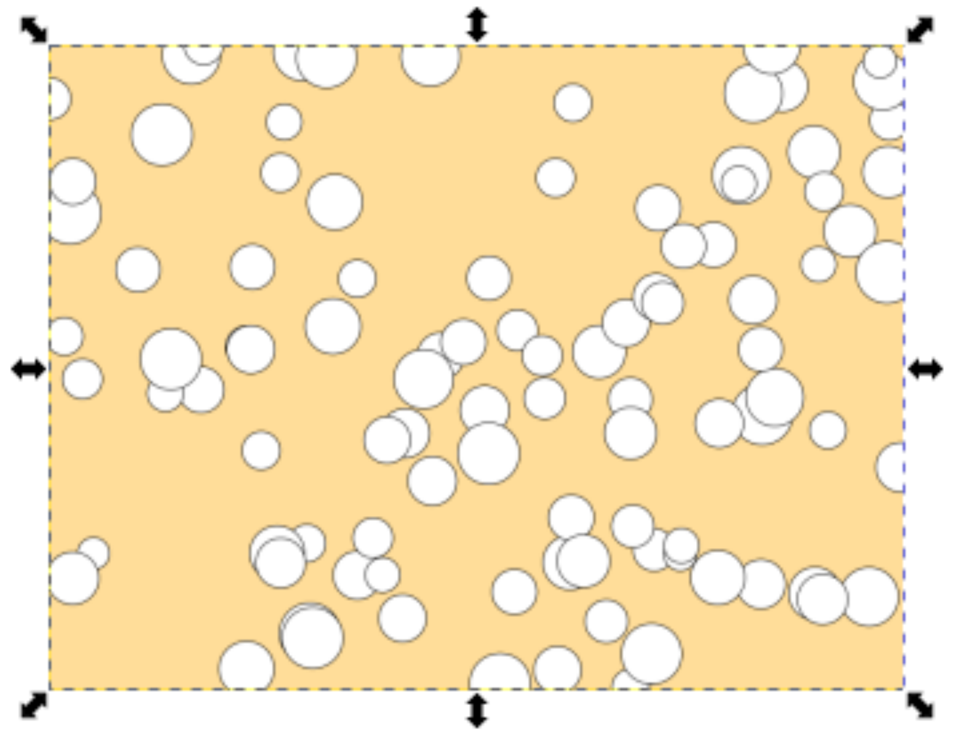
-elements

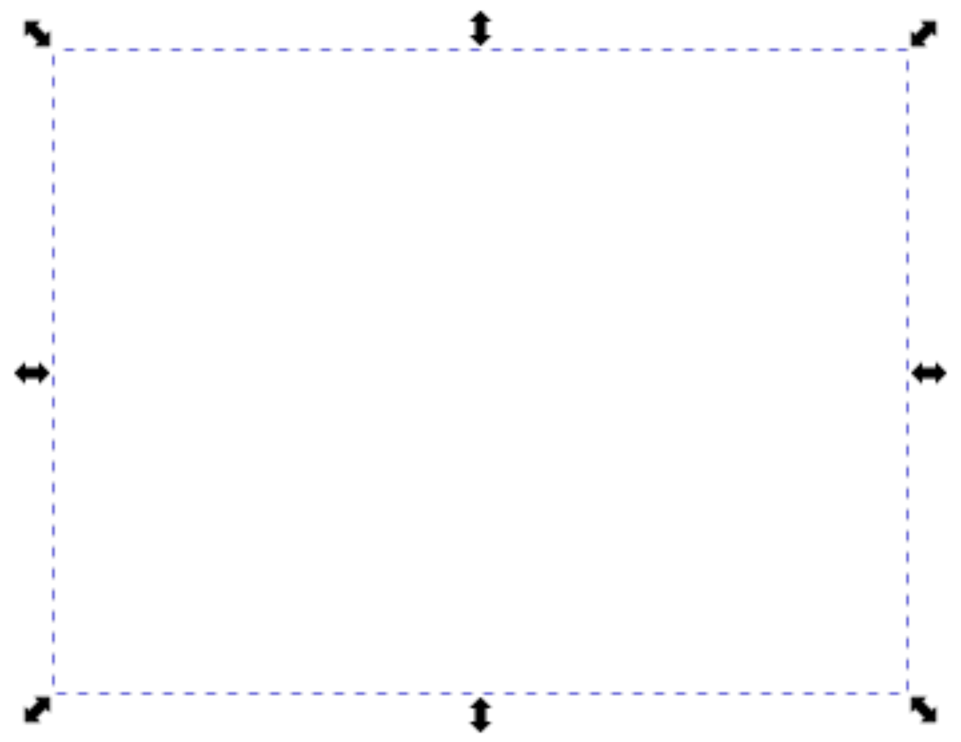
-attributes

“ SVG or Scalable Vector Graphics defines vector-based graphics in XML format ”

Jérôme Mutterer & Martin Owens

What Figure extensions do…





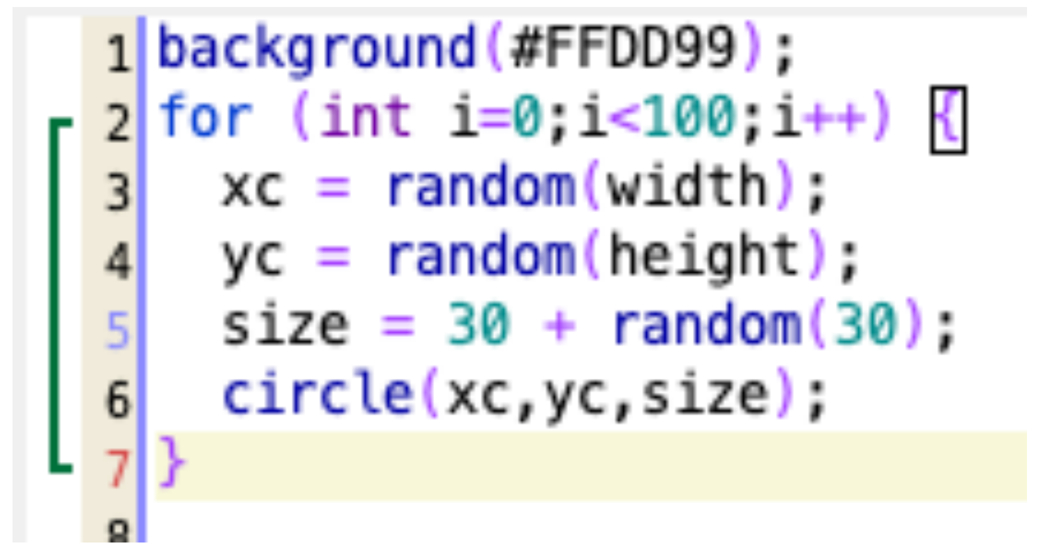
- x,y
- width
- height

script template

```
        script = f"""
// sketch
import processing.svg.*;
size({width},{height}, SVG, "{images_file}");

{description}

exit();
"""
```



```
background(#FFDD99);
for (int i=0;i<100;i++) {
  xc = random(width);
  yc = random(height);
  size = 30 + random(30);
  circle(xc,yc,size);
}
```

- code

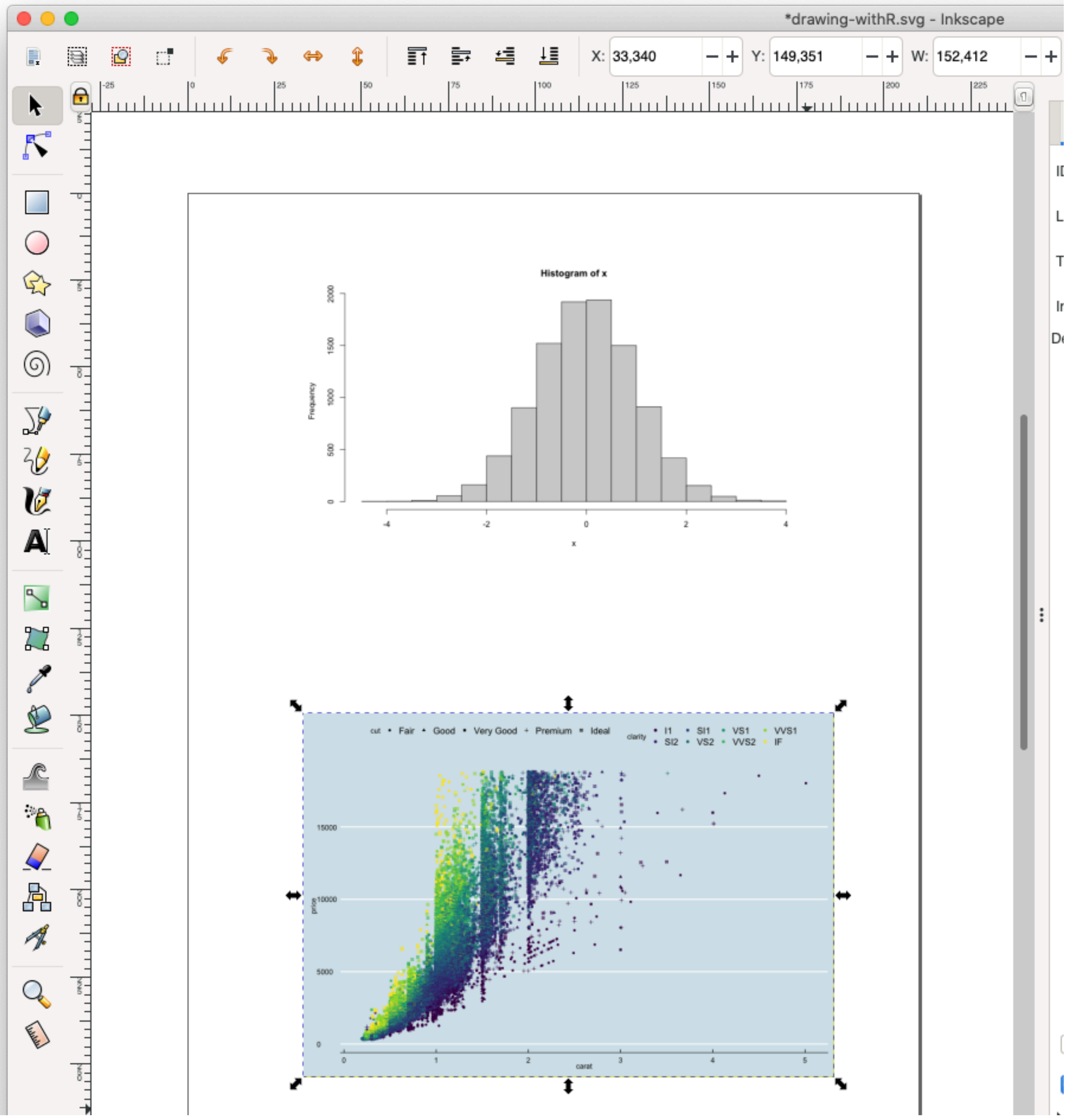
- Full script code file
- Panel image file
- Replace Rectangle by Image
- Insert image data
- Reinstate original description

```
>$ processing-java --sketch=<path> --run
```

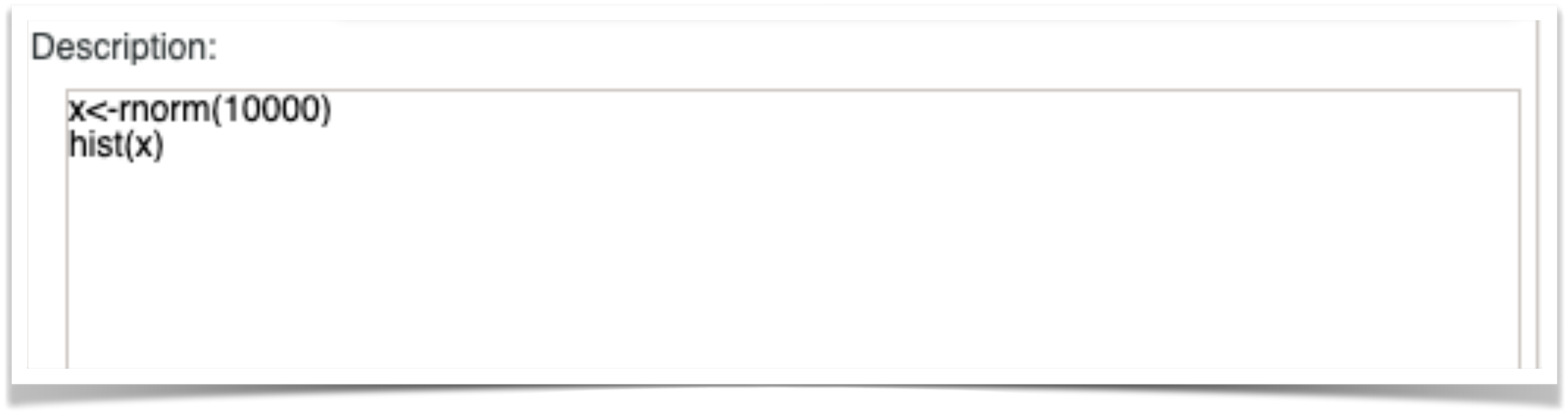
Jérôme Mutterer & Martin Owens

Graphs from R

the *view* is the output from a .R RScript

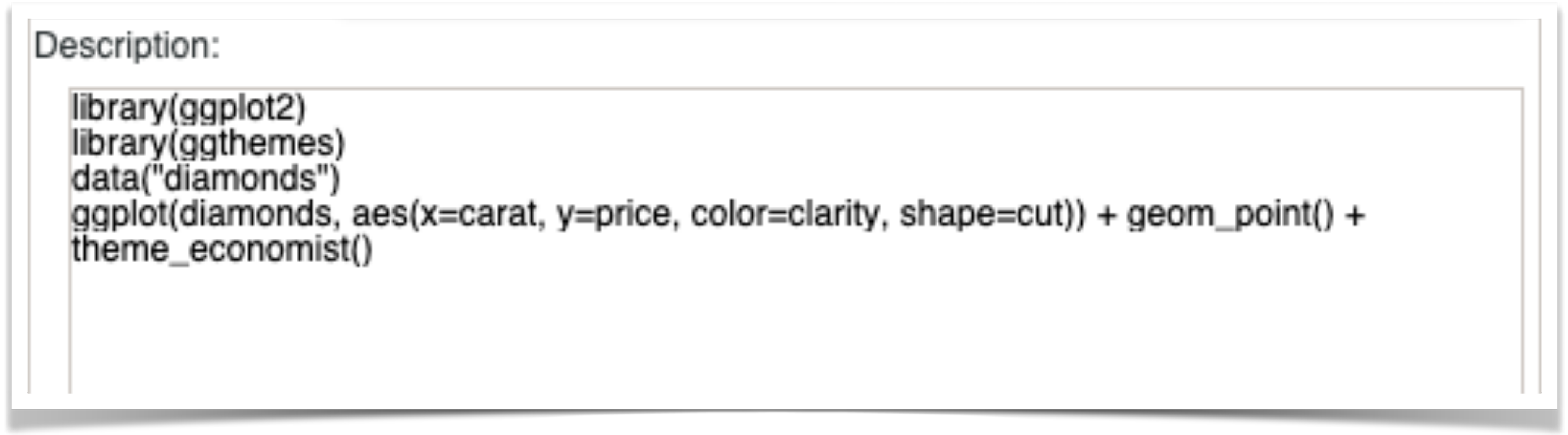


Simple R plot


Description:

```
x<-rnorm(10000)
hist(x)
```

Formatted ggplot

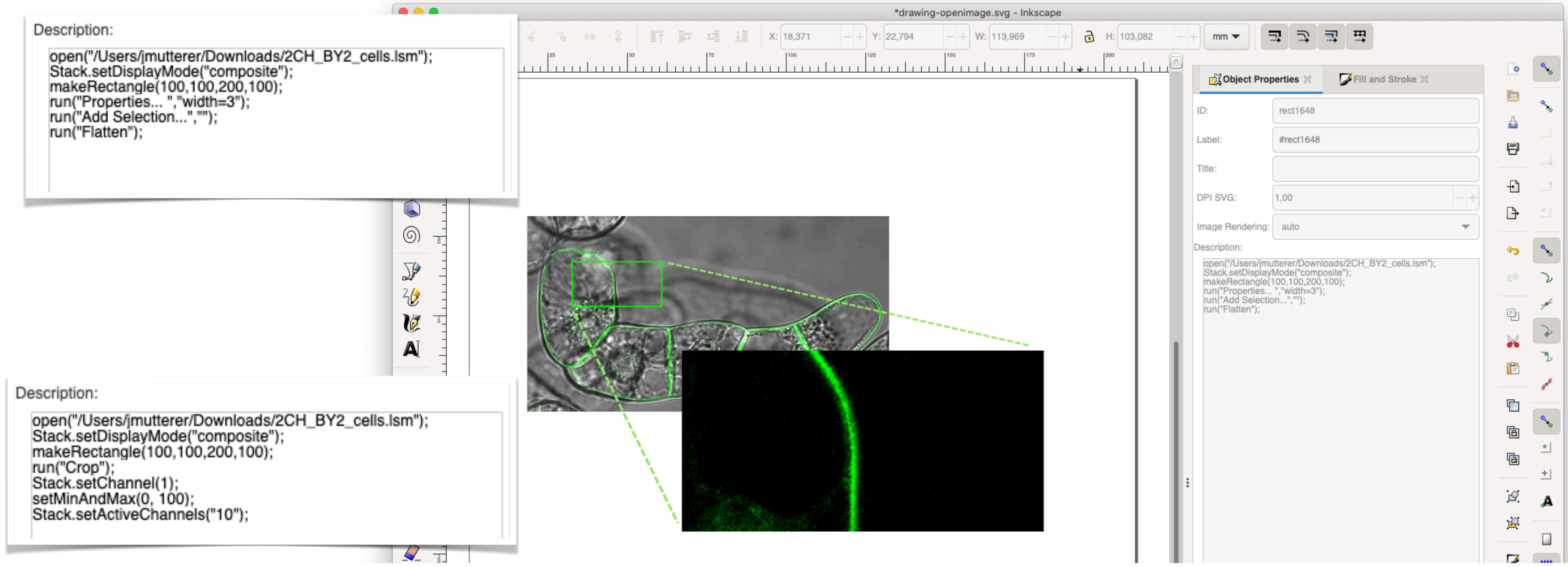

Description:

```
library(ggplot2)
library(ggthemes)
data("diamonds")
ggplot(diamonds, aes(x=carat, y=price, color=clarity, shape=cut)) + geom_point() +
theme_economist()
```

Jérôme Mutterer & Martin Owens

Images from ImageJ macros

the *view* is the output from macro



Description:

```
open("/Users/jmutterer/Downloads/2CH_BY2_cells.lsm");
Stack.setDisplayMode("composite");
makeRectangle(100,100,200,100);
run("Properties... ","width=3");
run("Add Selection...","");
run("Flatten");
```

X: 18,371 Y: 22,794 W: 113,969 H: 103,082 mm

Object Properties | Fill and Stroke

ID: rect1648

Label: #rect1648

Title:

DPI SVG: 1,00

Image Rendering: auto

Description:

```
open("/Users/jmutterer/Downloads/2CH_BY2_cells.lsm");
Stack.setDisplayMode("composite");
makeRectangle(100,100,200,100);
run("Properties... ","width=3");
run("Add Selection...","");
run("Flatten");
```

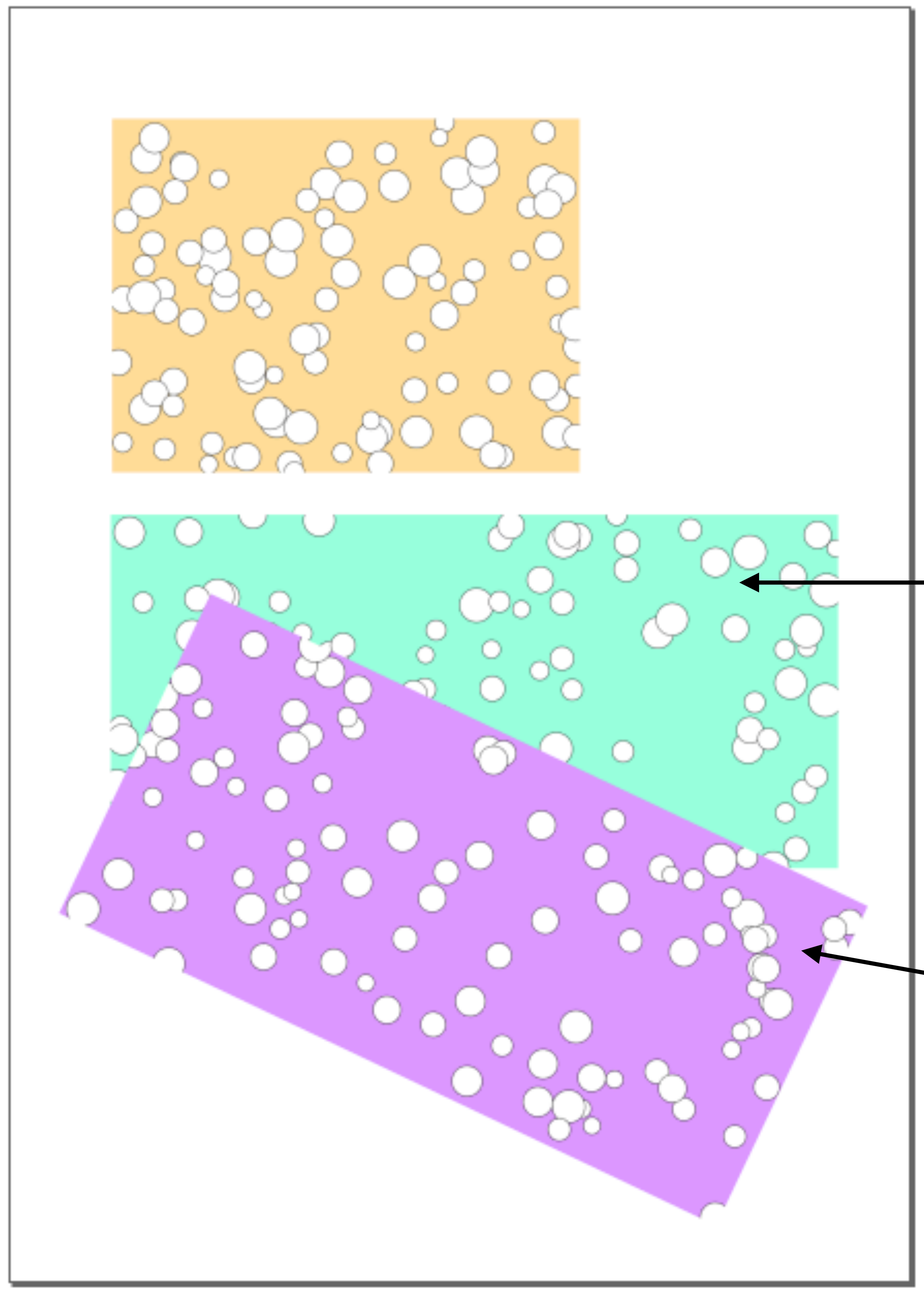
Description:

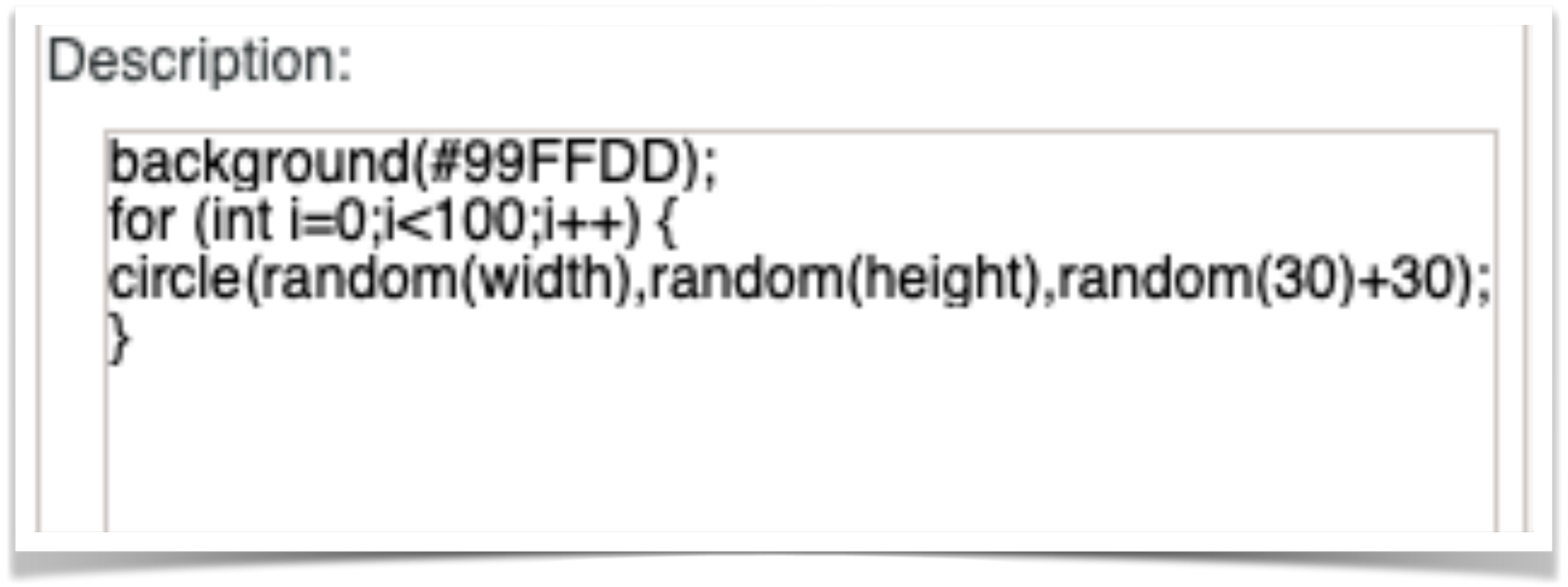
```
open("/Users/jmutterer/Downloads/2CH_BY2_cells.lsm");
Stack.setDisplayMode("composite");
makeRectangle(100,100,200,100);
run("Crop");
Stack.setChannel(1);
setMinAndMax(0, 100);
Stack.setActiveChannels("10");
```

Jérôme Mutterer & Martin Owens

Designs using Processing

example extension following the same principle





Description:

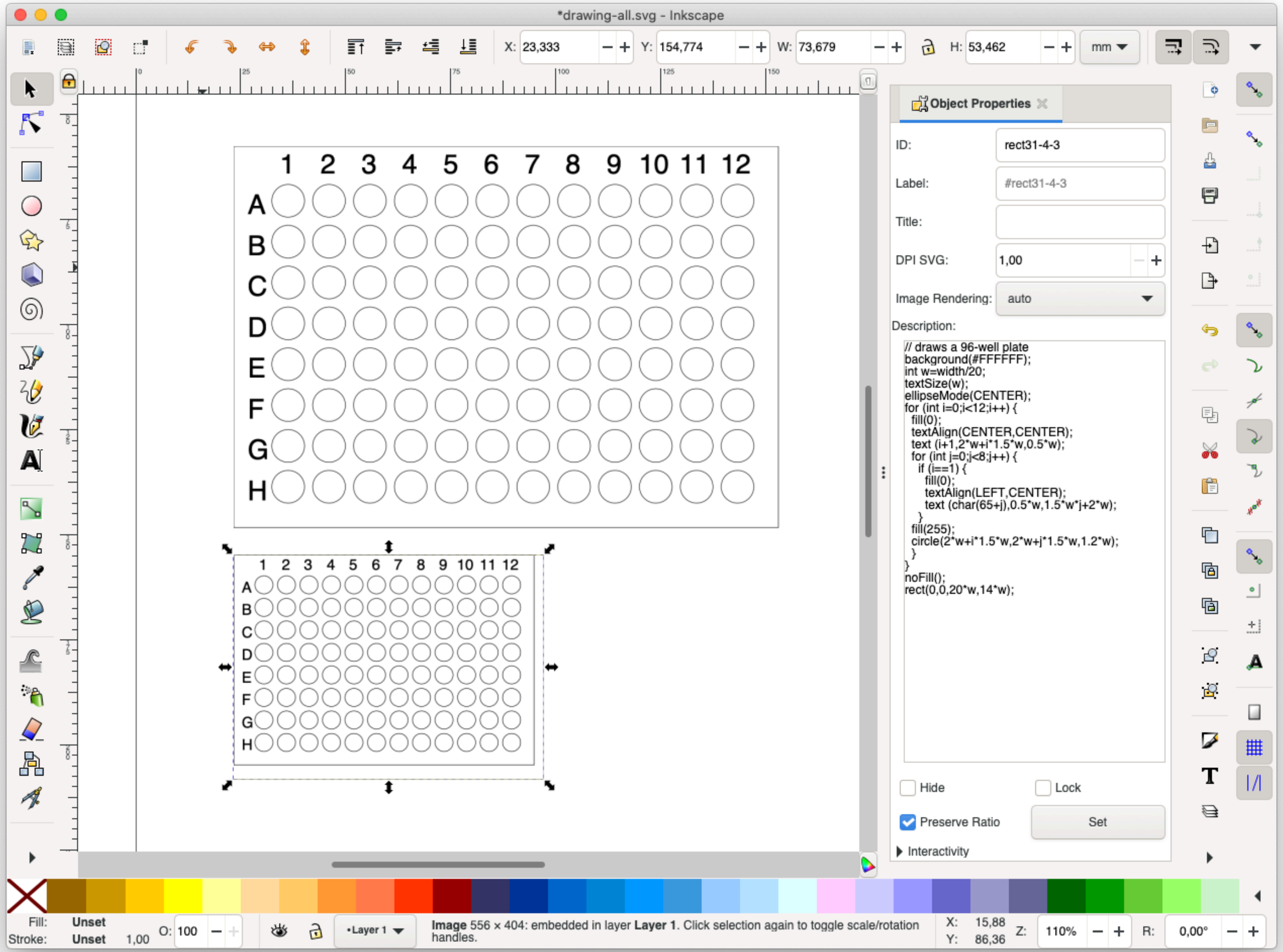
```
background(#99FFDD);
for (int i=0;i<100;i++) {
circle(random(width),random(height),random(30)+30);
}
```

Additional transform applied at svg level by Inkscape

Jérôme Mutterer & Martin Owens

Designs using Processing

svg with runnable code description as *widgets*



*drawing-all.svg - Inkscape

X: 23,333 Y: 154,774 W: 73,679 H: 53,462 mm

Object Properties

ID: rect31-4-3

Label: #rect31-4-3

Title:

DPI SVG: 1,00

Image Rendering: auto

Description:

```
// draws a 96-well plate
background(#FFFFFF);
int w=width/20;
textSize(w);
ellipseMode(CENTER);
for (int i=0;i<12;i++) {
  fill(0);
  textAlign(CENTER,CENTER);
  text (i+1,2*w+i*1.5*w,0.5*w);
  for (int j=0;j<8;j++) {
   if (i==1) {
     fill(0);
     textAlign(LEFT,CENTER);
     text (char(65+j),0.5*w,1.5*w*j+2*w);
   }
  fill(255);
  circle(2*w+i*1.5*w,2*w+j*1.5*w,1.2*w);
  }
}
noFill();
rect(0,0,20*w,14*w);
```

Hide Lock

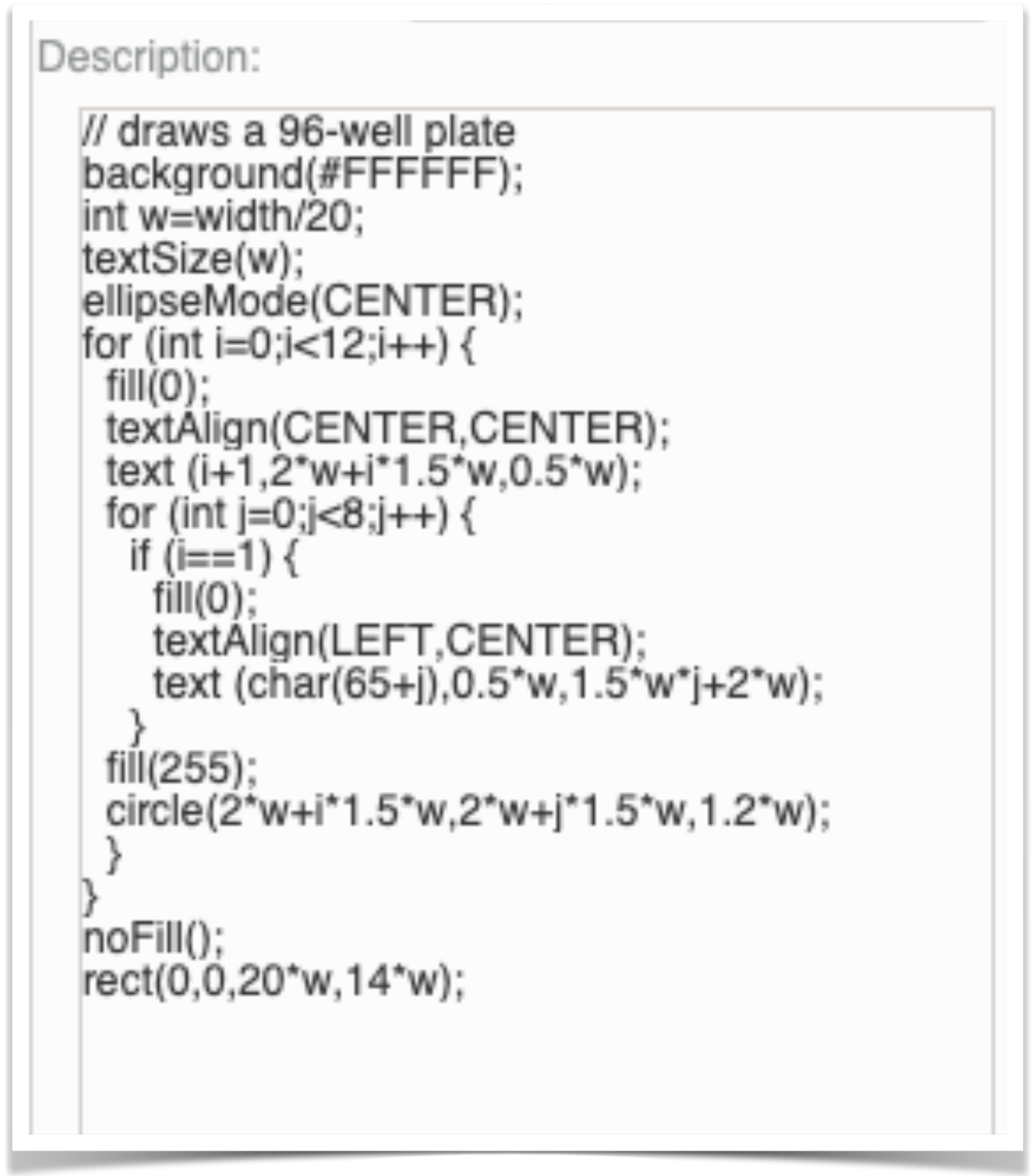
Preserve Ratio [✓] Set

Interactivity

Fill: Unset Stroke: Unset 1,00 O: 100 •Layer 1

Image 556 × 404: embedded in layer **Layer 1**. Click selection again to toggle scale/rotation handles.

X: 15,88 Y: 86,36 Z: 110% R: 0,00°



Description:

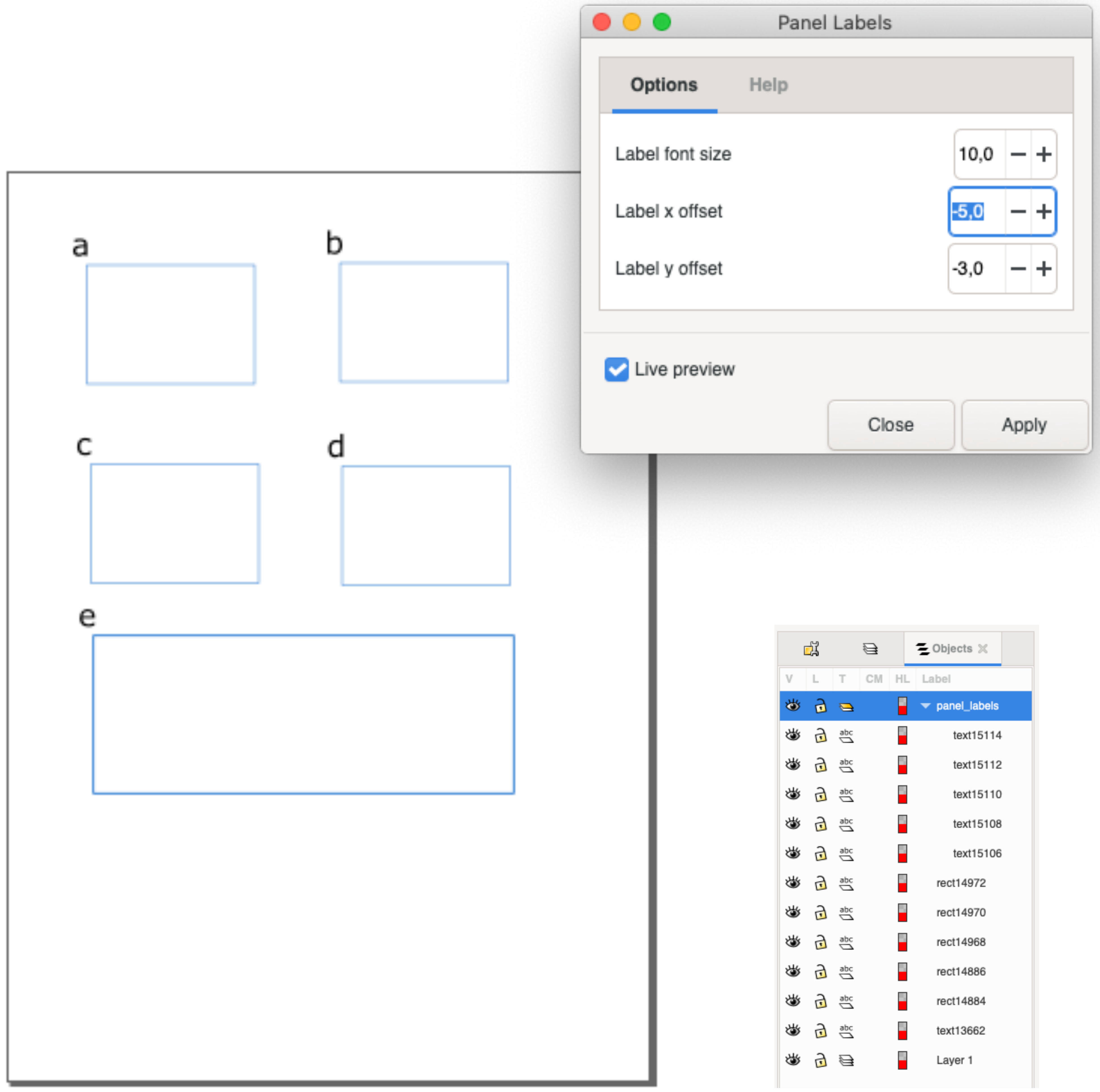
```
// draws a 96-well plate
background(#FFFFFF);
int w=width/20;
textSize(w);
ellipseMode(CENTER);
for (int i=0;i<12;i++) {
  fill(0);
  textAlign(CENTER,CENTER);
  text (i+1,2*w+i*1.5*w,0.5*w);
  for (int j=0;j<8;j++) {
   if (i==1) {
     fill(0);
     textAlign(LEFT,CENTER);
     text (char(65+j),0.5*w,1.5*w*j+2*w);
   }
  fill(255);
  circle(2*w+i*1.5*w,2*w+j*1.5*w,1.2*w);
  }
}
noFill();
rect(0,0,20*w,14*w);
```

Jérôme Mutterer & Martin Owens

Additional useful extensions

Panel Labels

Lane Labels



Panel Labels

Options | Help

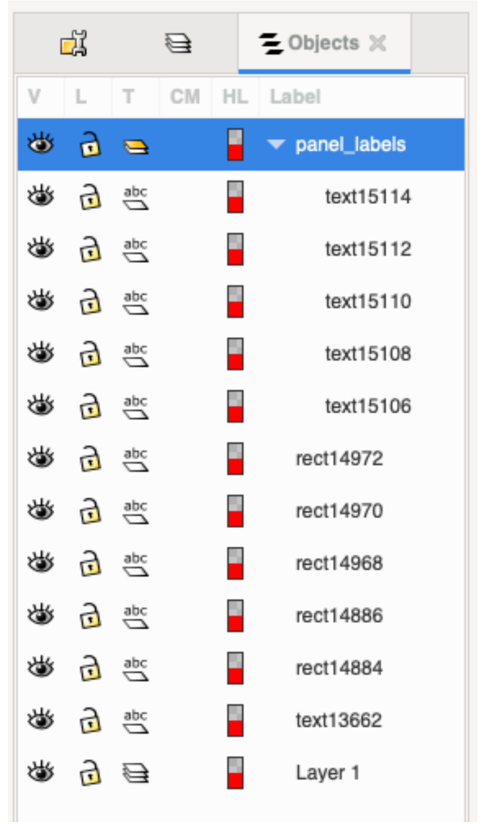
Label font size 10,0

Label x offset -5,0

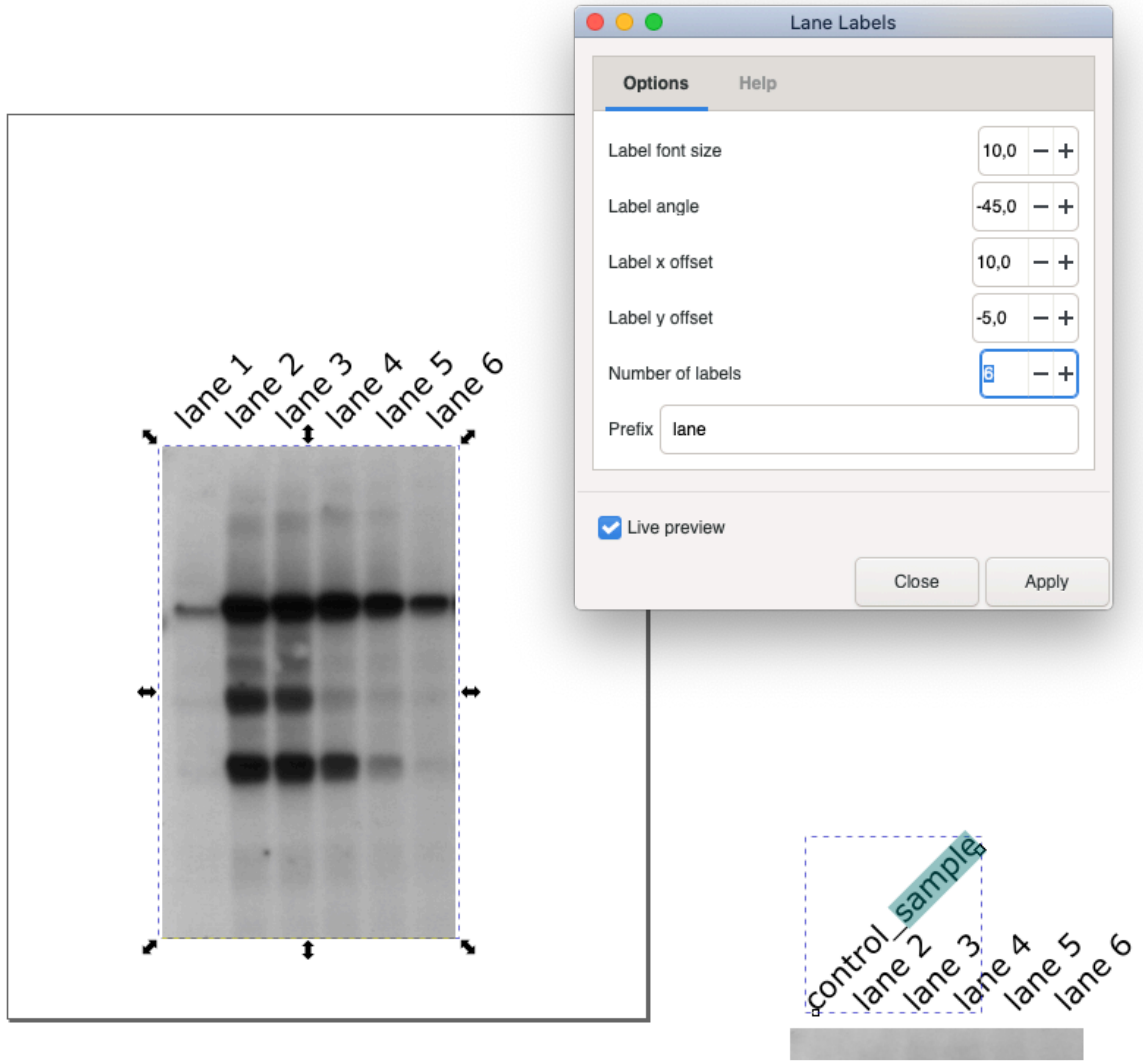
Label y offset -3,0

✓ Live preview

Close | Apply



V	L	T	CM	HL	Label
					panel_labels
					text15114
					text15112
					text15110
					text15108
					text15106
					rect14972
					rect14970
					rect14968
					rect14886
					rect14884
					text13662
					Layer 1



Lane Labels

Options | Help

Label font size 10,0

Label angle -45,0

Label x offset 10,0

Label y offset -5,0

Number of labels 6

Prefix lane

✓ Live preview

Close | Apply

Jérôme Mutterer & Martin Owens

Future directions

- Interoperability with more third party content providers.
- Creation of pure Inkscape extensions for assisted figure formatting.
- Enhancing user experience with a better object Properties dialog.
- Consider adding semi automated workflow.
- Setup detailed instructions for installation.
- Provide *svg templates*.
- Make use of Inkscape clipart library.
  (=create a library of *svg+code* with common *views* or graph types)

⚠ Requires Inkscape 1.1 or later