Image segmentation

1. Fertilised image of a sperm cell is processed with **Slic** and **Slico** algorithm using OpenImageR package.

In [3]:

```
library(OpenImageR)
```

In [18]:

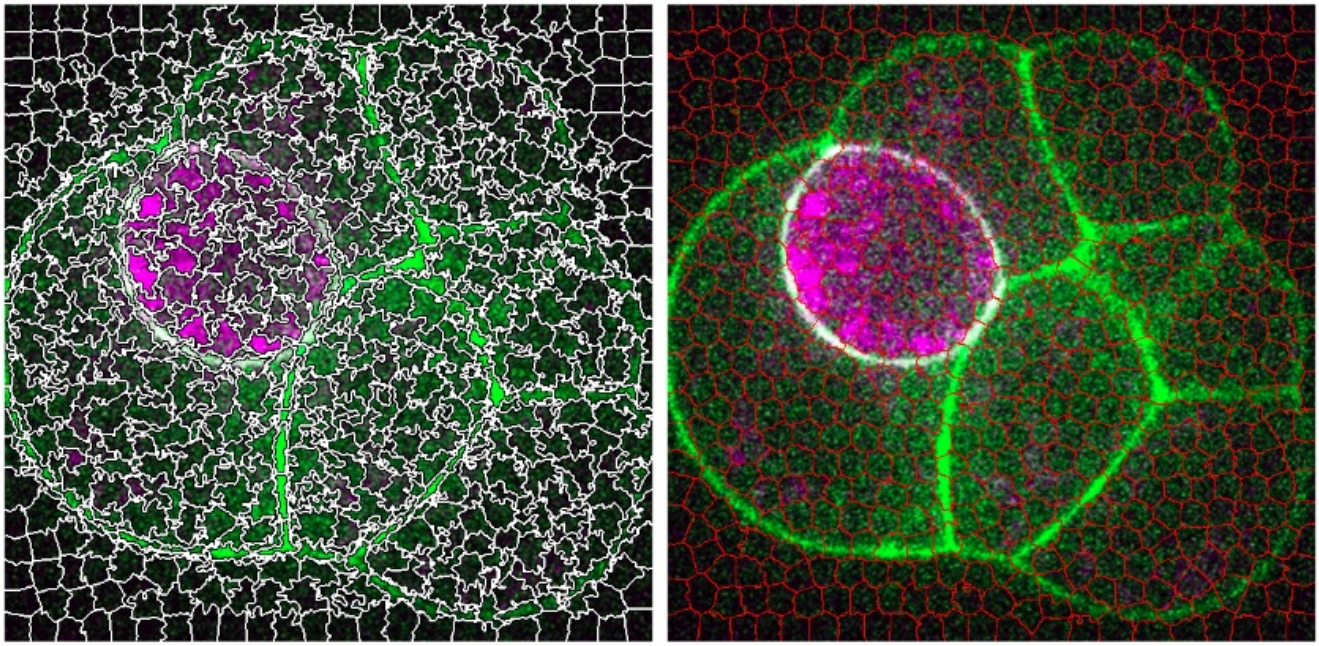
```
path = system.file("tmp_images", "c4.png", package = "OpenImageR")
im = readImage(path)
res_slic = superpixels(input_image = im,
                       method = "slic",
                       superpixel = 600,
                       compactness = 20,
                       return_slic_data = TRUE,
                       return_labels = TRUE,
                       write_slic = "",
                       verbose = TRUE)
str(res_slic)
res_slico = superpixels(input_image = im,
                        method = "slico",
                        superpixel = 600,
                        return_slic_data = TRUE,
                        return_labels = TRUE,
                        write_slic = "",
                        verbose = TRUE)
str(res_slico)
par(mfrow=c(1,2), mar = c(0.2, 0.2, 0.2, 0.2))

plot_slic = OpenImageR::NormalizeObject(res_slic$slic_data)
plot_slic = grDevices::as.raster(plot_slic)
graphics::plot(plot_slic)

plot_slico = OpenImageR::NormalizeObject(res_slico$slic_data)
plot_slico = grDevices::as.raster(plot_slico)
graphics::plot(plot_slico)
```

```
Warning message in interface_superpixels(input_image, method, superpixel, compactness, :
"The input data has more than 3 dimensions. The dimensions were reduced from 4 to 3!"Warning messa
ge in interface_superpixels(input_image, method, superpixel, compactness, :
"The input data has values between 0.000000 and 1.000000. The image-data will be multiplied by the
value: 255!"
```

```
The input image has the following dimensions: 584 594 3
The 'slic' method will be utilized!
The output image has the following dimensions: 584 594 3
List of 2
 $ slic_data: num [1:584, 1:594, 1:3] 14 15 13 13 13 14 12 11 11 10 ...
 $ labels   : num [1:584, 1:594] 0 0 0 0 0 0 0 0 0 0 ...
```

```
Warning message in interface_superpixels(input_image, method, superpixel, compactness, :
"The input data has more than 3 dimensions. The dimensions were reduced from 4 to 3!"Warning messa
ge in interface_superpixels(input_image, method, superpixel, compactness, :
"The input data has values between 0.000000 and 1.000000. The image-data will be multiplied by the
value: 255!"
```

```
The input image has the following dimensions: 584 594 3
The 'slico' method will be utilized!
The output image has the following dimensions: 584 594 3
List of 2
 $ slic_data: num [1:584, 1:594, 1:3] 14 15 13 13 13 14 12 11 11 10 ...
 $ labels   : num [1:584, 1:594] 0 0 0 0 0 0 0 0 0 0 ...
```



1. Image of fruit fly embryo

In [12]:

```
path = system.file("tmp_images", "c3.png", package = "OpenImageR")
im = readImage(path)
res_slic = superpixels(input_image = im,
                       method = "slic",
                       superpixel = 600,
                       compactness = 20,
                       return_slic_data = TRUE,
                       return_labels = TRUE,
                       write_slic = "",
                       verbose = TRUE)
str(res_slic)
res_slico = superpixels(input_image = im,
                        method = "slico",
                        superpixel = 600,
                        return_slic_data = TRUE,
                        return_labels = TRUE,
                        write_slic = "",
                        verbose = TRUE)
str(res_slico)
par(mfrow=c(1,2), mar = c(0.2, 0.2, 0.2, 0.2))

plot_slic = OpenImageR::NormalizeObject(res_slic$slic_data)
plot_slic = grDevices::as.raster(plot_slic)
graphics::plot(plot_slic)
```

```
plot_slico = OpenImageR::NormalizeObject(res_slico$slic_data)
plot_slico = grDevices::as.raster(plot_slico)
graphics::plot(plot_slico)
```

```
Warning message in interface_superpixels(input_image, method, superpixel, compactness, :
"The input data has more than 3 dimensions. The dimensions were reduced from 4 to 3!"Warning messa
ge in interface_superpixels(input_image, method, superpixel, compactness, :
"The input data has values between 0.000000 and 1.000000. The image-data will be multiplied by the
value: 255!"
```

```
The input image has the following dimensions: 460 588 3
The 'slic' method will be utilized!
The output image has the following dimensions: 460 588 3
List of 2
 $ slic_data: num [1:460, 1:588, 1:3] 0 0 0 4 15 0 7 6 1 0 ...
 $ labels   : num [1:460, 1:588] 0 0 0 0 0 0 0 0 0 0 ...
```

```
Warning message in interface_superpixels(input_image, method, superpixel, compactness, :
"The input data has more than 3 dimensions. The dimensions were reduced from 4 to 3!"Warning messa
ge in interface_superpixels(input_image, method, superpixel, compactness, :
"The input data has values between 0.000000 and 1.000000. The image-data will be multiplied by the
value: 255!"
```

```
The input image has the following dimensions: 460 588 3
The 'slico' method will be utilized!


Error in interface_superpixels(input_image, method, superpixel, compactness, : The 'K' parameter (
number-of-superpixels) should be bigger than 'n' in the 'SLICO::GetLABXYSeeds_ForGivenK()' functio
n!
Traceback:

1. superpixels(input_image = im, method = "slico", superpixel = 600,
 .     return_slic_data = TRUE, return_labels = TRUE, write_slic = "",
 .     verbose = TRUE)
2. interface_superpixels(input_image, method, superpixel, compactness,
 .     return_slic_data, return_lab_data, return_labels, write_slic,
 .     verbose)
```

In [17]:

```
path = system.file("tmp_images", "c3.png", package = "OpenImageR")
im = readImage(path)
res_slic = superpixels(input_image = im,
                       method = "slic",
                       superpixel = 300,
                       compactness = 20,
                       return_slic_data = TRUE,
                       return_labels = TRUE,
                       write_slic = "",
                       verbose = TRUE)
str(res_slic)
res_slico = superpixels(input_image = im,
                        method = "slico",
                        superpixel = 300,
                        return_slic_data = TRUE,
                        return_labels = TRUE,
                        write_slic = "",
                        verbose = TRUE)
str(res_slico)
par(mfrow=c(1,2), mar = c(0.2, 0.2, 0.2, 0.2))

plot_slic = OpenImageR::NormalizeObject(res_slic$slic_data)
plot_slic = grDevices::as.raster(plot_slic)
graphics::plot(plot_slic)

plot_slico = OpenImageR::NormalizeObject(res_slico$slic_data)
plot_slico = grDevices::as.raster(plot_slico)
graphics::plot(plot_slico)
```

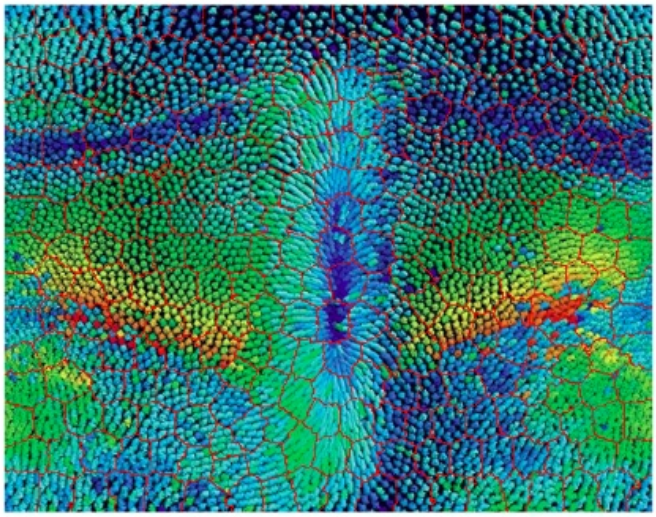
```
Warning message in interface_superpixels(input_image, method, superpixel, compactness, :
"The input data has more than 3 dimensions. The dimensions were reduced from 4 to 3!"Warning messa
```

```
ge in interface_superpixels(input_image, method, superpixel, compactness, :
“The input data has values between 0.000000 and 1.000000. The image-data will be multiplied by the value: 255!”
```

```
The input image has the following dimensions: 460 588 3
The 'slic' method will be utilized!
The output image has the following dimensions: 460 588 3
List of 2
 $ slic_data: num [1:460, 1:588, 1:3] 0 0 0 4 15 0 7 6 1 0 ...
 $ labels   : num [1:460, 1:588] 0 0 0 0 0 0 0 0 0 0 ...
```

```
Warning message in interface_superpixels(input_image, method, superpixel, compactness, :
“The input data has more than 3 dimensions. The dimensions were reduced from 4 to 3!”Warning messa
ge in interface_superpixels(input_image, method, superpixel, compactness, :
“The input data has values between 0.000000 and 1.000000. The image-data will be multiplied by the value: 255!”
```

```
The input image has the following dimensions: 460 588 3
The 'slico' method will be utilized!
The output image has the following dimensions: 460 588 3
List of 2
 $ slic_data: num [1:460, 1:588, 1:3] 0 0 0 4 15 0 7 6 1 0 ...
 $ labels   : num [1:460, 1:588] 0 0 0 0 0 0 0 0 0 0 ...
```



Conclusion:

Seeing the result of two image that has been processed it can be concluded that slico is better image segmentation algorithm than slic because boundaries of image with slico algoritm are more smooth and correctly recongnised.

The another benefits is that compactness is not required to be given manually in slico algorithm and it is automatically decided by program based on texture and smoothness of the image.

Since we are filling compactness manually in slic algorithm that make it uniform compact parameter throurghout image irrespective of texture and smoothness.