

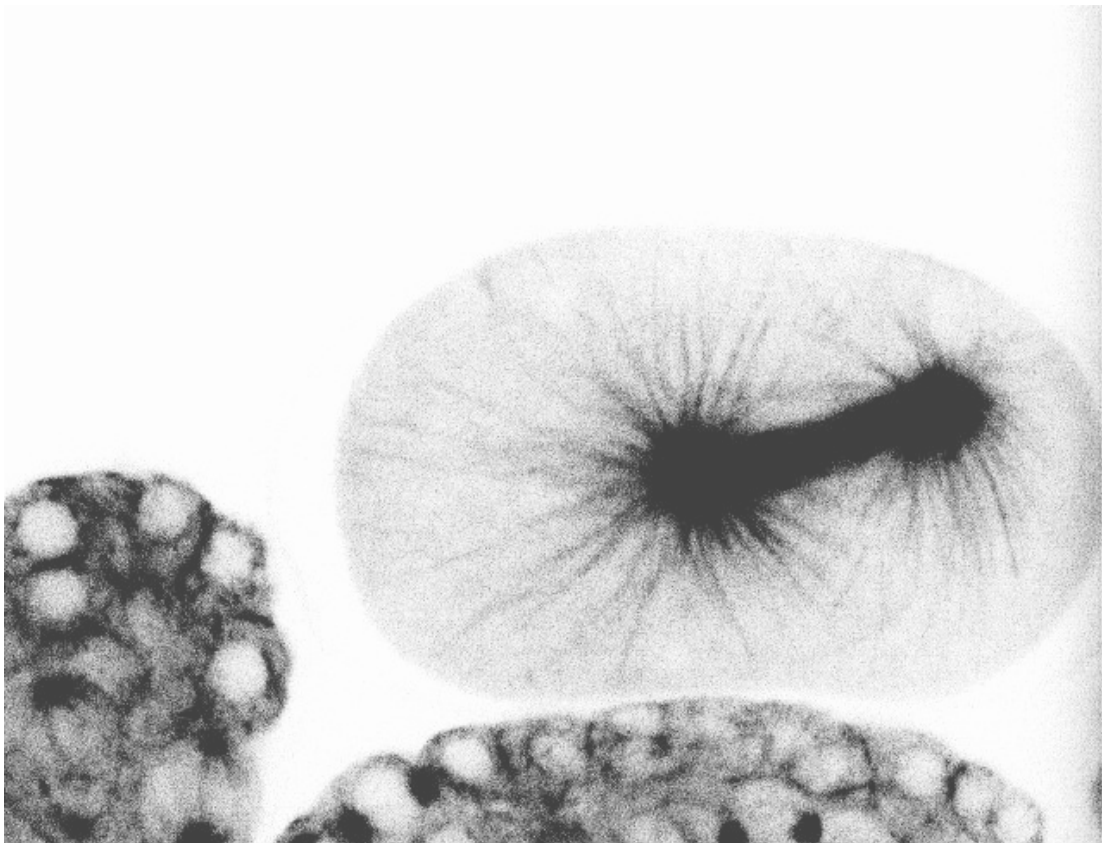
Questão 1

Apenas explicativa.

Questão 2.1

```
static public ImageAccess inverse(ImageAccess input) {  
    int nx = input.getWidth();  
    int ny = input.getHeight();  
    ImageAccess output = new ImageAccess(nx, ny);  
    double value = 0.0;  
    for (int x=0; x<nx; x++)  
        for (int y=0; y<ny; y++) {  
            value = input.getPixel(x, y);  
            value = 255 - value;  
            output.putPixel(x, y, value);  
        }  
    return output;  
}
```

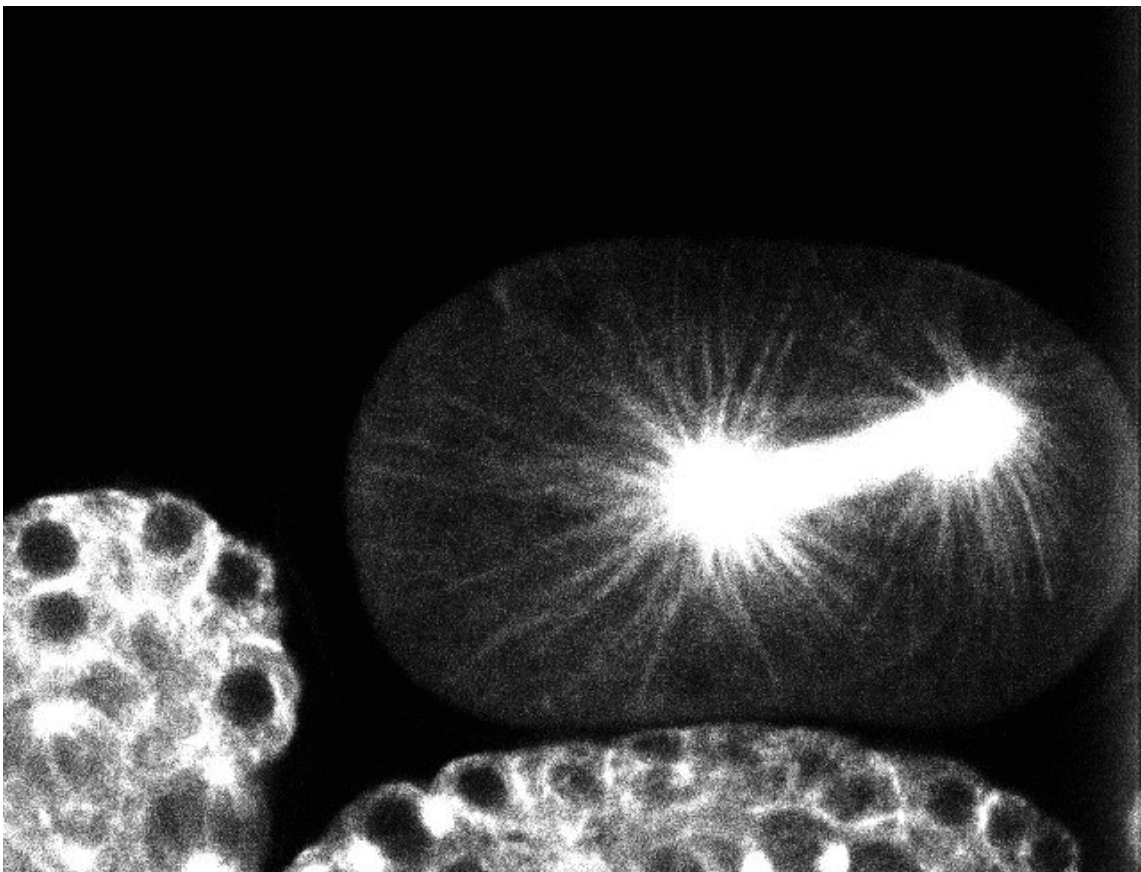
microtubules.tif + inverse:



Questão 2.2

```
static public ImageAccess rescale(ImageAccess input) {  
    int nx = input.getWidth();  
    int ny = input.getHeight();  
    double max = input.getMaximum();  
    double min = input.getMinimum();  
    ImageAccess output = new ImageAccess(nx, ny);  
  
    double alpha = 255 / (max - min);  
    double beta = min;  
  
    double value = 0.0;  
    for (int x=0; x<nx; x++)  
        for (int y=0; y<ny; y++) {  
            value = input.getPixel(x, y);  
            value = alpha * (value - beta);  
            output.putPixel(x, y, value);  
        }  
  
    return output;  
}
```

microtubules.tif + rescale:



Questão 2.3

```
static public ImageAccess saturate(ImageAccess input) {  
    int nx = input.getWidth();  
    int ny = input.getHeight();  
    ImageAccess output = new ImageAccess(nx, ny);  
  
    double value = 0.0;  
    for (int x=0; x<nx; x++)  
        for (int y=0; y<ny; y++) {  
            value = input.getPixel(x, y);  
  
            if(value > 10000) {  
                value = 10000;  
            }  
  
            output.putPixel(x, y, value);  
        }  
  
    return output;  
}
```

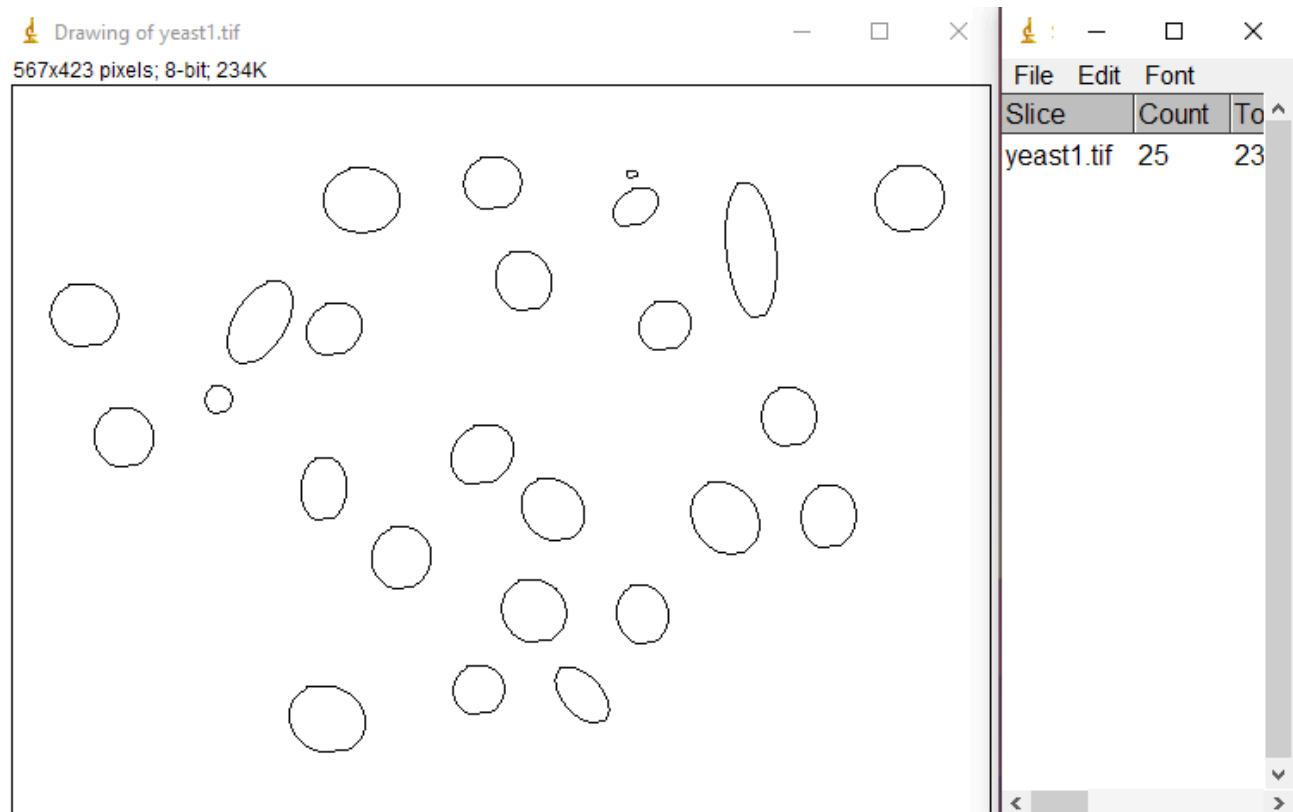
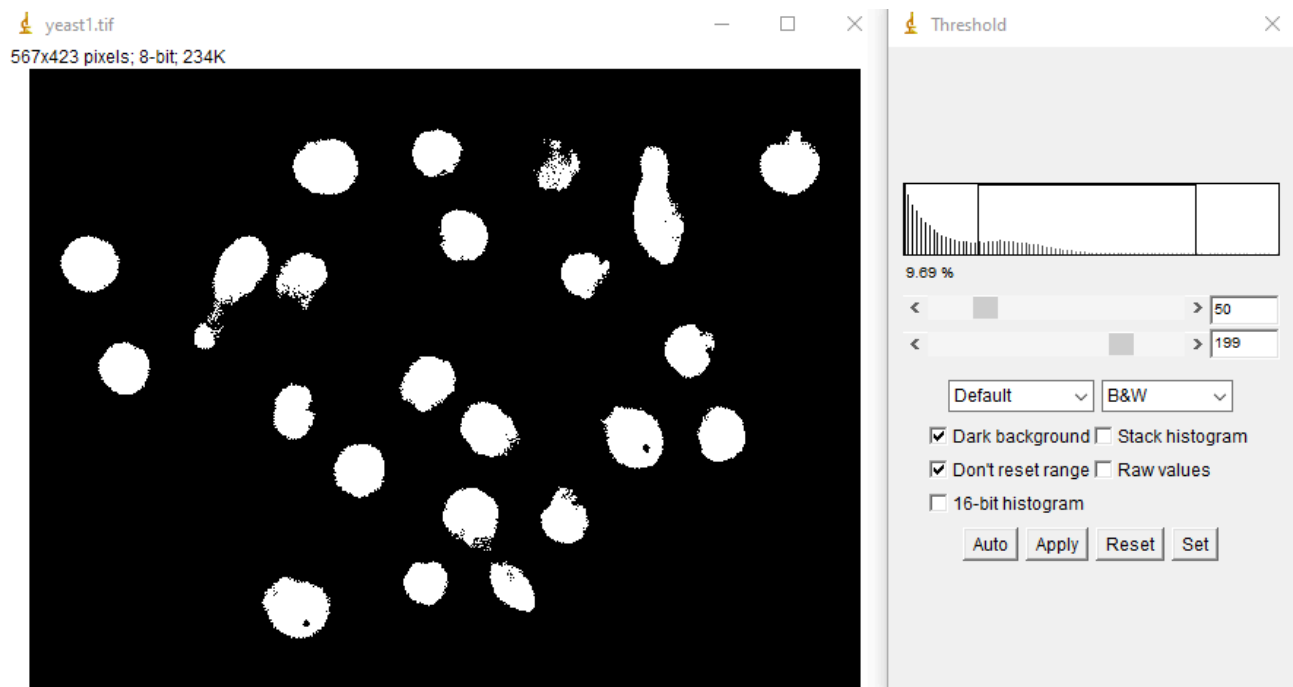
HRTC.tif + saturate + rescale:



Questão 3.1

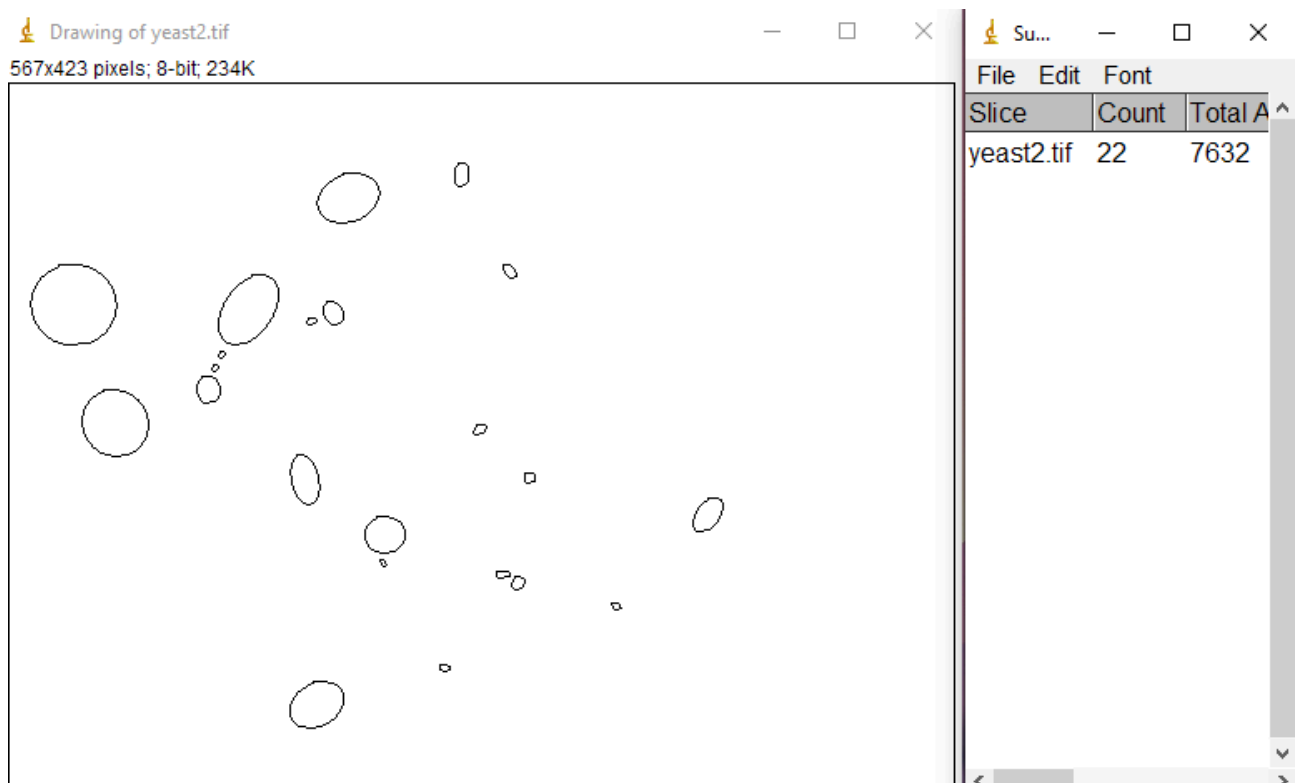
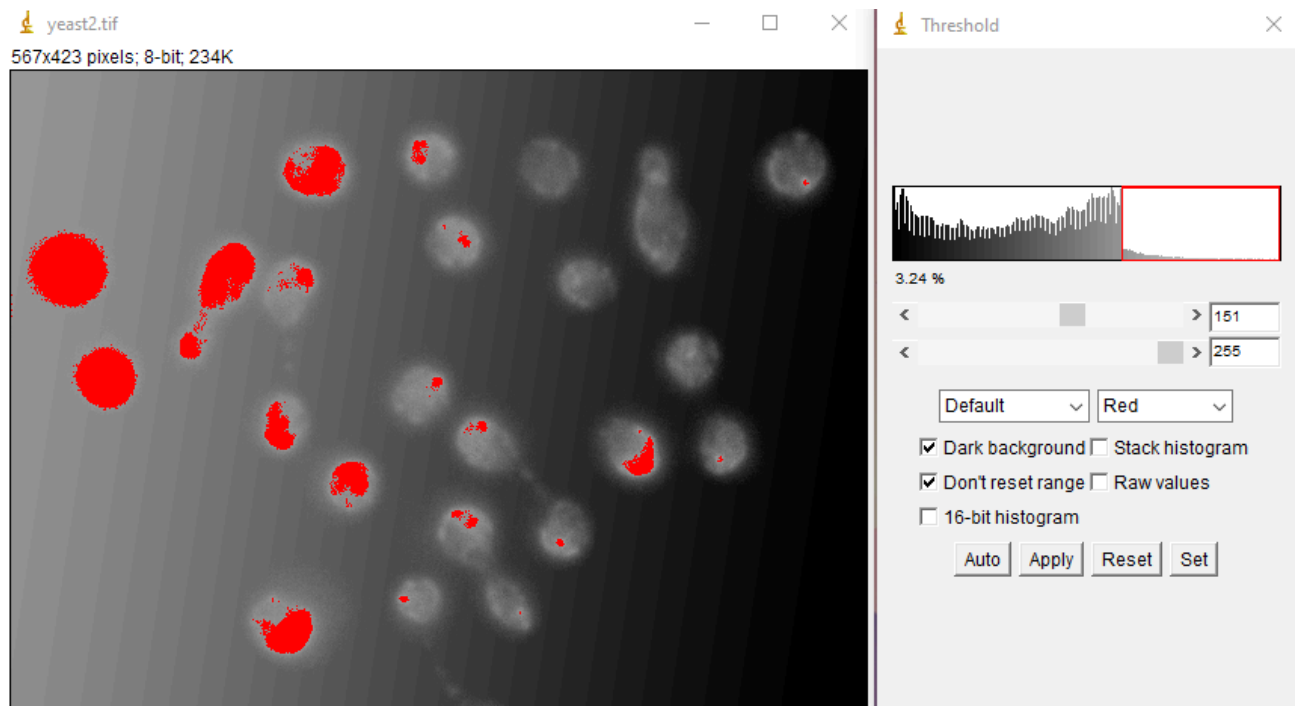
O threshold do analisador de partículas foi calibrado em 10-infinity pixel units para todas as imagens a fim de evitar detecções vazias. O threshold do contraste/brilho foi ajustado individualmente.

yeast1:

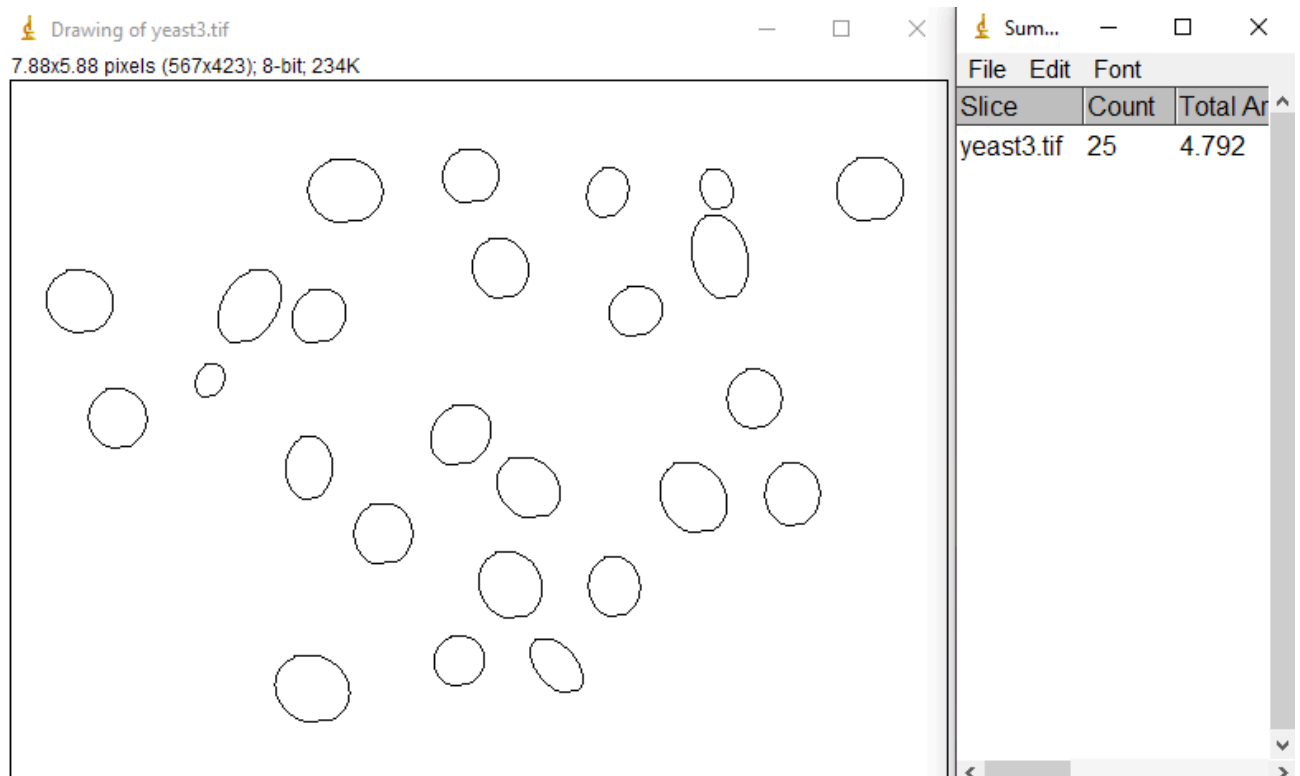
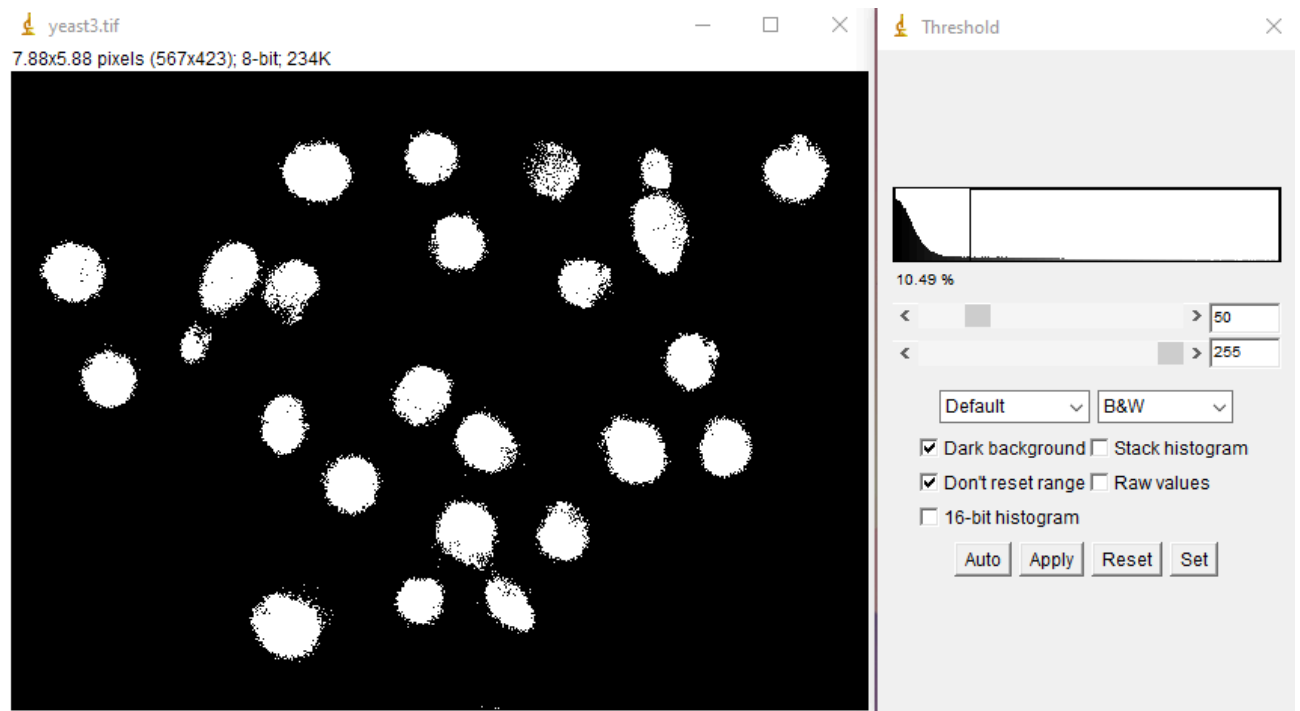


yeast2:

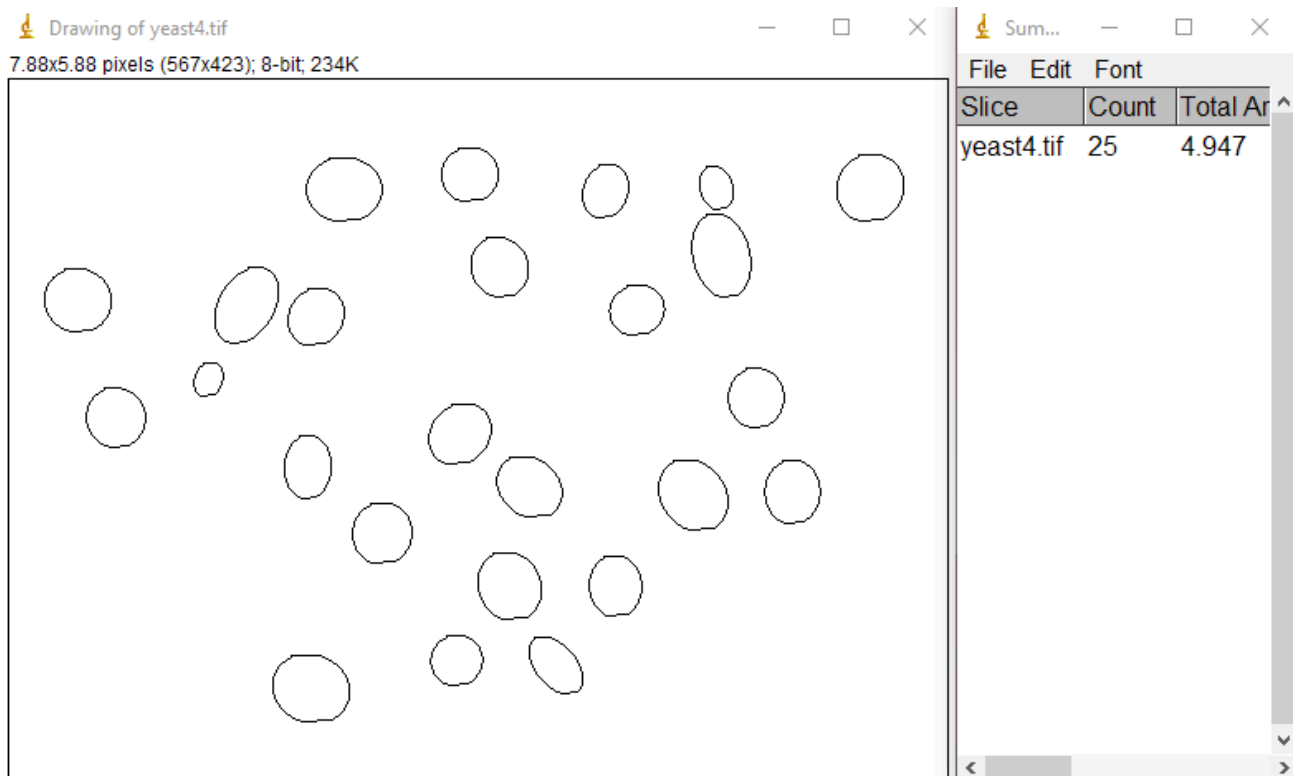
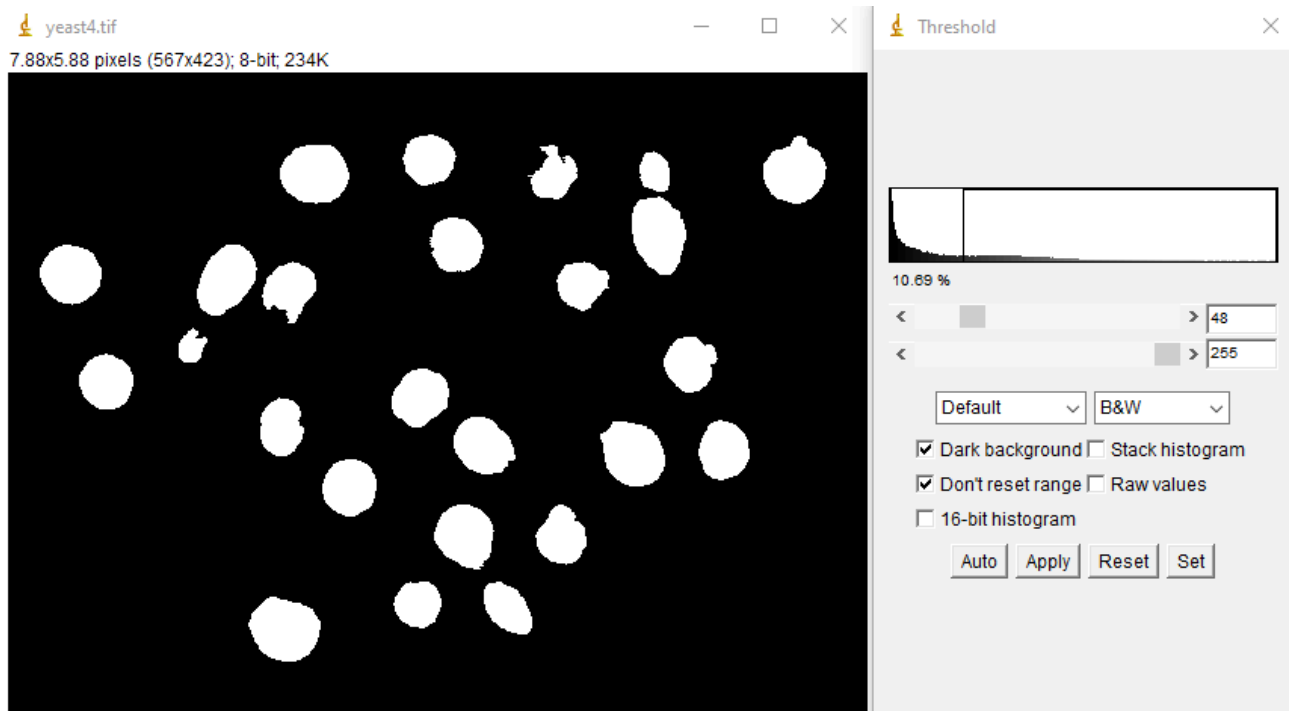
Aqui é possível observar que é impossível definir um threshold que abarque todas as 25 imagens. A diferença dos níveis do ruído de um lado da imagem são mais altos do que a própria representação relevante no outro lado da imagem.



yeast3:



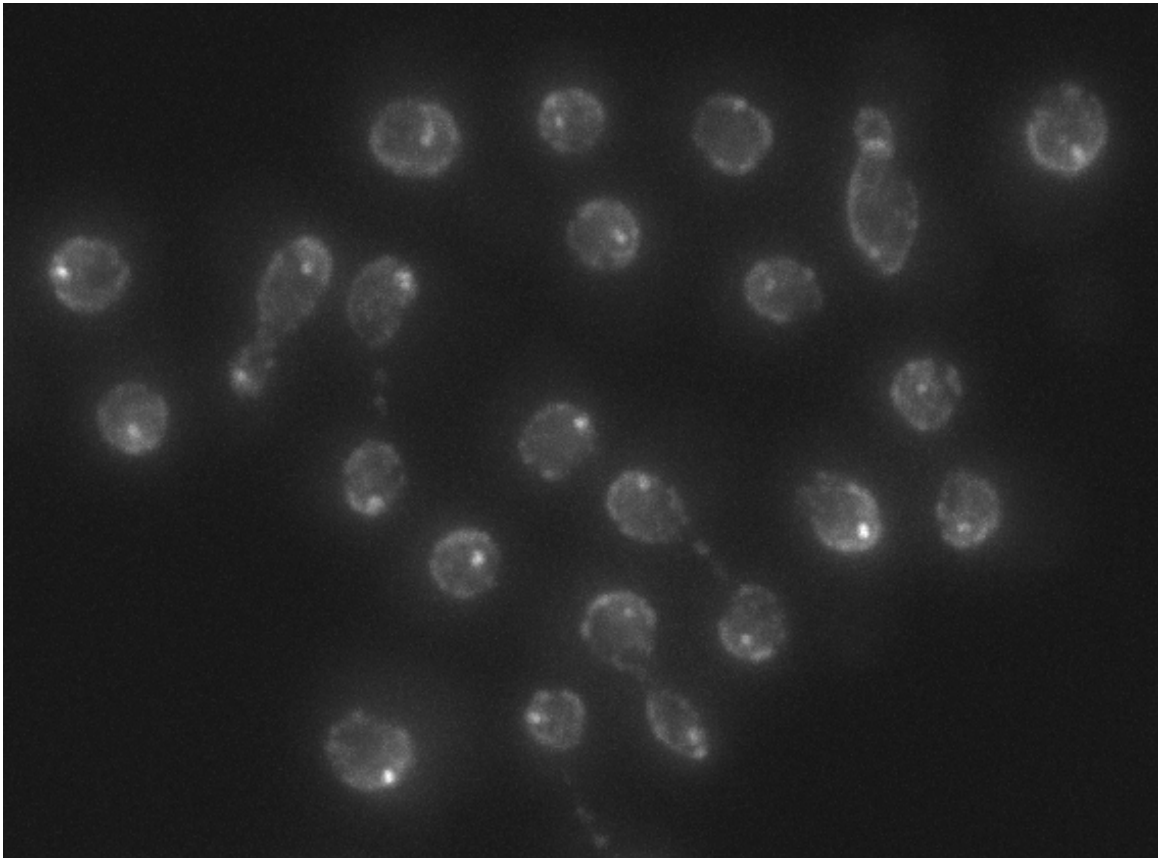
yeast4:



Questão 4.1

```
static public ImageAccess zprojectMaximum(ImageAccess[] zstack) {  
    int nx = zstack[0].getWidth();  
    int ny = zstack[0].getHeight();  
    int nz = zstack.length;  
    ImageAccess output = new ImageAccess(nx, ny);  
  
    double value = 0.0;  
  
    for (int x=0; x<nx; x++)  
        for (int y=0; y<ny; y++) {  
            value = zstack[0].getPixel(x, y);  
            for (int z=1; z<nz; z++) {  
                if (zstack[z].getPixel(x, y) > value) {  
                    value = zstack[z].getPixel(x, y);  
                }  
            }  
            output.putPixel(x, y, value);  
        }  
  
    return output;  
}
```

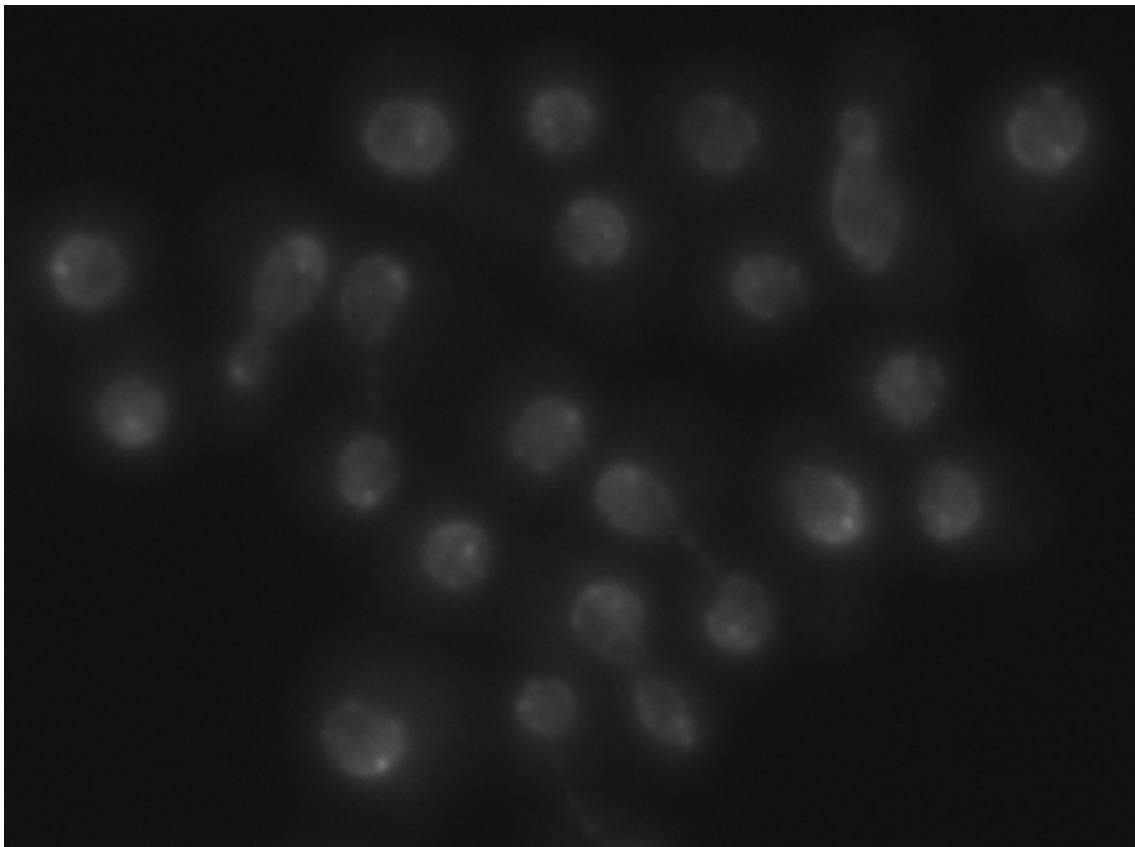
yeard_stack.tif + zprojectMaximum:



Questão 4.2

```
static public ImageAccess zprojectMean(ImageAccess[] zstack) {  
    int nx = zstack[0].getWidth();  
    int ny = zstack[0].getHeight();  
    int nz = zstack.length;  
    ImageAccess output = new ImageAccess(nx, ny);  
  
    double value = 0.0;  
  
    for (int x=0; x<nx; x++)  
    for (int y=0; y<ny; y++) {  
        value = 0.0;  
        for (int z=0; z<nz; z++) {  
            value += zstack[z].getPixel(x, y);  
        }  
        value /= nz;  
        output.putPixel(x, y, value);  
    }  
  
    return output;  
}
```

yeast_stack.tif + zprojectMean:



Questão 4.3

yeast_stack.tif + brightness/contrast + merge(yeast_phase.tif):

