Example furrow segmentation

Giorgia Guglielmi, Joseph D. Barry, Wolfgang Huber, Stefano De Renzis

1 Introduction

In this vignette we show an example furrow segmentation using image data from the paper by Guglielmi *et al*.

2 Load Data

We first load an example movie of a wild-type (without photoactivation or any other perturbation) furrowing embryo.

```
data("exampleFurrowMovie")
img <- exampleFurrowMovie
rm(exampleFurrowMovie)
```

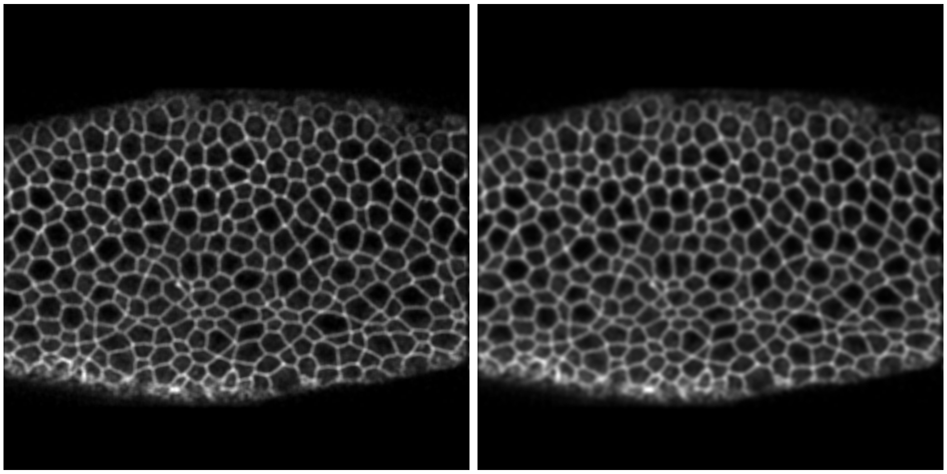
3 Step by step segmentation

We first set the segmentation parameters. For descriptions of the parameters see the help for the main segmentation function `?segmentFurrowAllStacks`.

```
threshOffset <- 0.0005
px <- 0.293
filterSize <- makeOdd(round(microns2px(1, px=px)))
L <- makeOdd(round(microns2px(5, px=px)))
minObjectSize <- area2px(4, px=px)
maxObjectSize <- area2px(400, px=px)
```

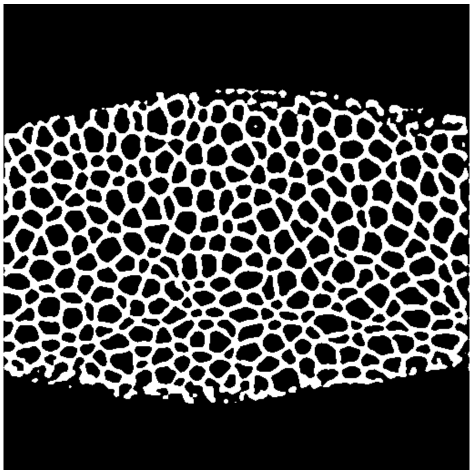
To reduce pixel noise the images were smoothed with a Gaussian filter.

```
z <- makeBrush(size=filterSize, shape="gaussian", sigma=filterSize/2)
display(normalize(img[, , 1, 100]), method="raster")
img2 <- filter2(img, z)
display(normalize(img2[, , 1, 100]), method="raster")
```



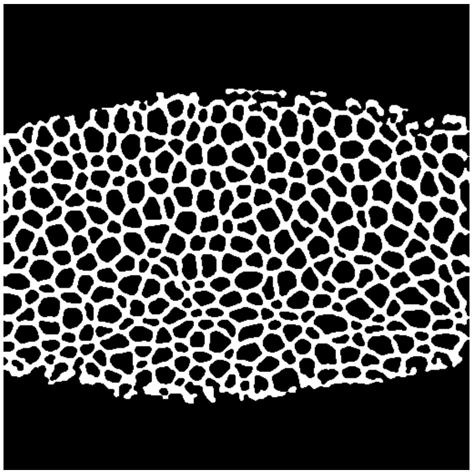
Adaptive thresholding was performed to coarsely identify membrane regions. A more accurate membrane identification will be obtained later in this vignette.

```r
mask <- thresh(x=img2[, , 1, ], w=L/2, h=L/2, offset=threshOffset)
display(mask[, , 100], method="raster")
```



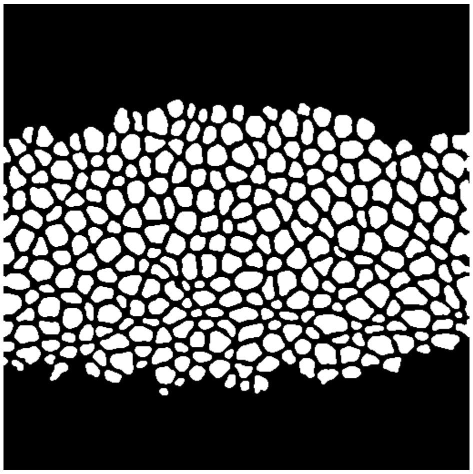
A closing morphological operation was performed, and very small objects were removed to smooth the mask further.

```r
brush <- makeBrush(size=filterSize, shape="disc")
mask <- closing(mask, brush)
mask <- bwlabel(mask)
mask <- furrowSeg:::filterObjects(mask, minObjectSize, Inf)
display(mask[, , 100], method="raster")
```



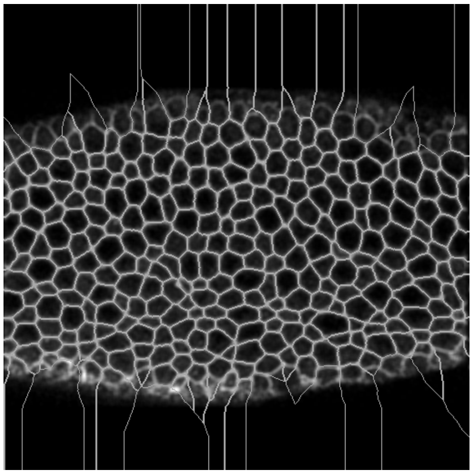
The mask was then inverted to obtain seed areas for cell nuclei. Object masks with holes were filled and overly large masks were removed since these were also unlikely to be nuclei.

```
mask <- furrowSeg:::invertMask(mask)
mask <- fillHull(mask)
mask <- furrowSeg:::filterObjects(mask, 0, maxObjectSize)
display(mask[, , 100], method="raster")
```



Next a propagate algorithm was run to accurately identify cell boundaries. The algorithm finds the voronoi region around each seed nucleus, with a distance metric that is a function of local image properties (see `?propagate`). We determined local image properties from the gaussian-smoothed image. Overly small and large candidate cells were again removed.

```
mask <- reenumerate(mask)
mask <- propagate(img2[, , 1, ], seeds=mask)
mask <- furrowSeg:::filterObjects(mask, minObjectSize, maxObjectSize)
hs <- paintObjects(x=mask, tgt=normalize(img), col="yellow")
display(hs[, , 1, 100], method="raster")
```



There were some remaining inaccurate segmentations around the edge of the embryo. For our paper we focussed on subsets of cells along the furrowing line, and so the inaccurate segmentations were not included.

The above steps were combined into a single function for easier computation. Finally we verify that the above steps produce identical results to the segmentation function.

```
x <- segmentFurrowAllStacks(x=img, L=L, filterSize=filterSize,
    threshOffset=threshOffset, closingSize=filterSize,
    minObjectSize=minObjectSize, maxObjectSize=maxObjectSize)

## Thresholding stack1

## Closing stack1

## Label objects stack1

## Remove objects stack1

## Invert mask stack1

## Fill holes stack1

## Remove objects 2 stack1

## Reenumerate cell ids stack1

## Propagate stack1

## Remove objects 3 stack1

## Paint objects stack1

all(mask == x$mask[[1]])

## [1] TRUE
```

4 Feature extraction

The following extracts image features using *EBImage*.

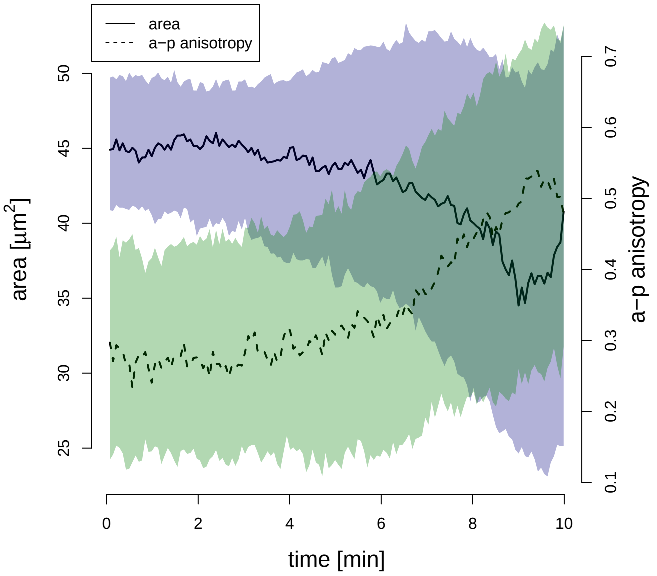
```
nt <- dim(mask)[3]
getEBImageFeatures <- function(mask, ref) {
    xbw <- reenumerate(mask)
    fts <- computeFeatures(x=xbw, ref=ref,
        methods.noref=c("computeFeatures.moment", "computeFeatures.shape"))
    return(fts)
}
ftList <- lapply(1:nt, function(t) {
    df <- getEBImageFeatures(mask[, , t], img[, , 1, t])
    df <- as.data.frame(df, stringsAsFactors=FALSE)
    df$t <- t
    return(df)
})
fts <- do.call("rbind", ftList)
```

The A-P (Anterior-Posterior) anisotropy measure `e.x` was calculated using the following. Note that this step requires that the image is aligned so that the A-P axis of the embryo is horizontal (parallel to the x-axis). An image rotation may therefore be required before performing the anisotropy calculation.

```
fts$e.x <- cos(fts$x.0.m.theta)*fts$x.0.m.eccentricity
```

As a simple example of how to visualize feature evolution, here we isolate cells in a rectangular box and plot A-P anisotropy and area over time. The timestep `dt` and pixel side-length `px` are specified in minutes and microns respectively.

```
box <- c("xleft"=64, "xright"=448, "ybottom"=128, "ytop"=384)
fts <- isolateBoxCells(fts, box)
plotFeatureEvolution(fts, tMax=10, dt=4.22/60, px=0.293)
```



For additional examples of how to process image feature data please see the accompanying vignette in this package, `vignette("genPaperFigures", package="furrowSeg")`.