

# MaxContrastProjection

*Jan Sauer*

October 30, 2018

## Contents

1	Getting Started . . . . .	1
2	Introduction . . . . .	1
3	Contrast Projection . . . . .	3
4	Application Example . . . . .	4
5	Session Info . . . . .	6

## 1 Getting Started

---

*MaxContrastProjection* is an *R* package to project 3D image stacks including common projections, such as the maximum and minimum intensity projections, as well as a novel maximum contrast projection to retain information about fine structures that would otherwise be lost in intensity- based projections. This package is distributed as part of the *Bioconductor* project and can be installed by starting *R* and running:

```
if (!requireNamespace("BiocManager", quietly=TRUE))
  install.packages("BiocManager")
BiocManager::install("MaxContrastProjection")
```

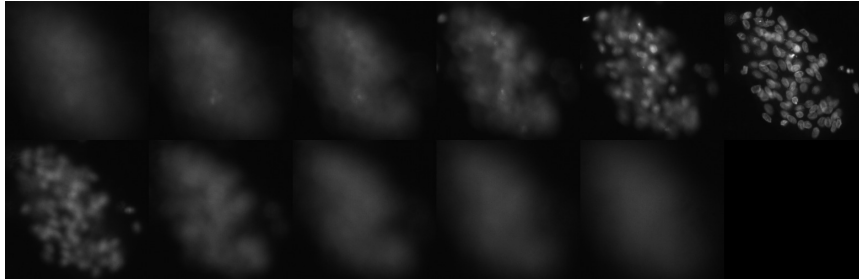
## 2 Introduction

---

A common problem when recording 3D fluorescent microscopy images is how to properly present these results in 2D. Large structures with elements in multiple focal planes become very difficult to image properly without the use of some form of projection. Objects in the focal plane will be sharp and the emitted light focused, whereas objects outside of the focal plane will be blurred and the emitted light is spread out over a larger area of the detector. Consequently, a maximum intensity projection is commonly used on the image stack in order to find the focal plane of an object.

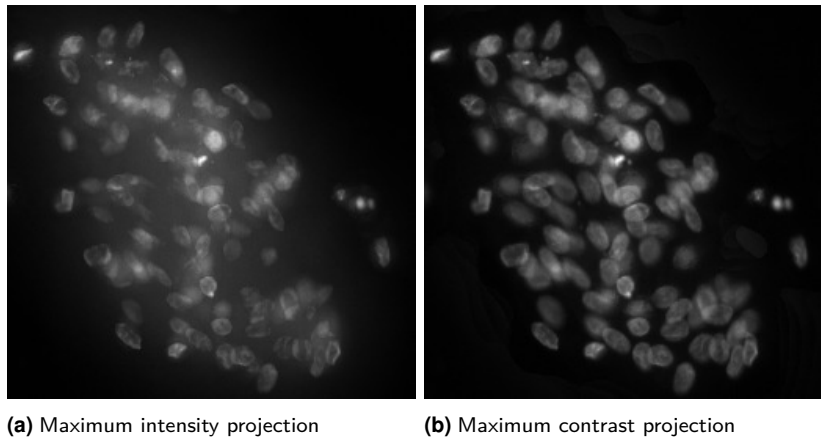
## MaxContrastProjection

The problem with this approach is that the blurring makes it more difficult to tell apart foreground and background of the image. If the exact position of an object in a medium is unknown, or if it has a 3D structure, it must be imaged at various focal lengths. Fig. 1 shows a segment of an organoid imaged at different focal lengths.



**Figure 1:** A cluster of cells in an organoid at different focal lengths

A look at the maximum intensity projection, shown in Fig. 2a, reveals that much of the distinction between foreground and background is lost in this projection. Consequently, a different type of projection is necessary to retain important information. The z-layer in which a region of an image is in focus is the z-layer with the least amount of blurring, i.e. with the highest contrast of intensity values within that area. Fig. 2b shows this "maximum contrast projection". Comparing it to the maximum intensity projection shows that the individual cells are much easier to differentiate.



**(a)** Maximum intensity projection

**(b)** Maximum contrast projection

**Figure 2:** A comparison of the maximum intensity projection and the maximum contrast projection of the images seen in Fig

1

### 3 Contrast Projection

The contrast  $C_{xy}$  at a point  $(x, y)$  is defined here as the variance of all intensity scores in a defined area around this point. In the simplest case, this is a rectangle centered around  $(x, y)$  with side lengths of  $2 \cdot w_x + 1$  and  $2 \cdot w_y + 1$ .

$$C_{xy} = \frac{1}{(2 \cdot w_x + 1) \cdot (2 \cdot w_y + 1)} \sum_{k=x-w_x}^{x+w_x} \sum_{l=y-w_y}^{y+w_y} (I_{kl} - \langle I_{xy} \rangle)^2 \quad 1$$

$$C_{xy} = \frac{1}{A} \sum_{k=x-w_x}^{x+w_x} \sum_{l=y-w_y}^{y+w_y} I_{kl}^2 - 2 \cdot I_{kl} \cdot \langle I_{xy} \rangle + \langle I_{xy} \rangle^2 \quad 2$$

$$C_{xy} = \langle I_{xy} \rangle^2 - 2 \cdot \langle I_{xy} \rangle \cdot \frac{1}{A} \sum_{k=x-w_x}^{x+w_x} \sum_{l=y-w_y}^{y+w_y} I_{kl} + \frac{1}{A} \sum_{k=x-w_x}^{x+w_x} \sum_{l=y-w_y}^{y+w_y} I_{kl}^2 \quad 3$$

where  $\langle I_{xy} \rangle$  is the intensity of the image at the position  $(x, y)$  and  $\langle I_{xy} \rangle$  is the mean intensity of the window centered around  $(x, y)$ :

$$\begin{aligned} \langle I_{xy} \rangle &= \frac{1}{(2 \cdot w_x + 1) \cdot (2 \cdot w_y + 1)} \sum_{k=x-w_x}^{x+w_x} \sum_{l=y-w_y}^{y+w_y} I_{kl} \\ &= \frac{1}{A} \sum_{k=x-w_x}^{x+w_x} \sum_{l=y-w_y}^{y+w_y} I_{kl} \end{aligned} \quad 4$$

This mean intensity over a window can be quickly determined in R with the function `EBImage::filter2()` by applying a normalized, uniform filter to the image. For brevity, the normalizing constant, i.e. the area of the window, is abbreviated as  $A = (2 \cdot w_x + 1) \cdot (2 \cdot w_y + 1)$ .

A comparison of Eqs. 4 and 3 shows that the variance takes on the known form  $Var = E[X^2] - (E[X])^2$ :

$$\begin{aligned} C_{xy} &= \langle I_{xy} \rangle^2 + \frac{1}{A} \sum_{k=x-w_x}^{x+w_x} \sum_{l=y-w_y}^{y+w_y} I_{kl}^2 - 2 \cdot \langle I_{xy} \rangle^2 \\ &= \frac{1}{A} \sum_{k=x-w_x}^{x+w_x} \sum_{l=y-w_y}^{y+w_y} I_{kl}^2 - \langle I_{xy} \rangle^2 = \langle I_{xy}^2 \rangle - \langle I_{xy} \rangle^2 \end{aligned} \quad 5$$

where the first term of Eq. 5 is the mean of the squared intensities over the same window as used for the mean of the intensities,  $\langle I_{xy}^2 \rangle = \frac{1}{A} \sum_{k=x-w_x}^{x+w_x} \sum_{l=y-w_y}^{y+w_y} I_{kl}^2$ . This quantity can be calculated by the same method as for the intensity using `EBImage::filter2()`

## 4 Application Example

We will use the image data for the organoids already presented in the Introduction of this vignette. In the most general case, the data should be an array representing a stack of images, i.e. the first two dimensions correspond to the  $x$ - and  $y$ - dimensions of the image and the third dimension of the array corresponds to the number of images in the stack.

The *cells* dataset represents a stack of 11 images, each with a size of 261 × 251 pixels (see Fig. 1)

```
library(MaxContrastProjection)
library(EBImage)
data(cells)
dim(cells)

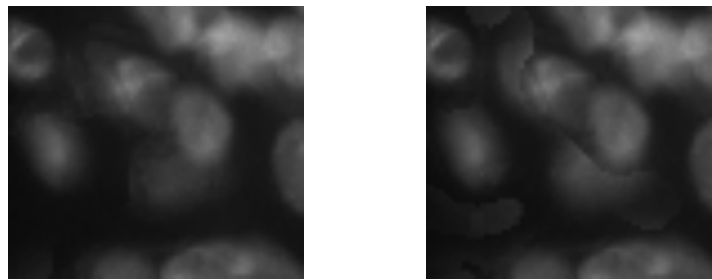
## [1] 261 251 11
```

The maximum contrast projection returns a single image with the same  $x$ - and  $y$ - dimensions as the input image stack

```
max_contrast = contrastProjection(imageStack = cells, w_x = 15, w_y = 15,
                                  smoothing = 5, brushShape = "box")
```

$w_x$  and  $w_y$  are the radii of the area around each pixel over which to calculate the variance. The area of this window should be at least as large as the structures of interest in the image, otherwise artefacts may occur in the projection. Figs. 3a and 3b show the impact of the window size on the projection quality. If the parameter  $w_y$  is not explicitly set, it is assumed to be equal to  $w_x$ .

```
max_contrast_large = contrastProjection(imageStack = cells[30:95, 55:115, ],
                                       w_x = 15, w_y = 15, smoothing = 5,
                                       brushShape = "box")
max_contrast_small = contrastProjection(imageStack = cells[30:95, 55:115, ],
                                       w_x = 3, w_y = 3, smoothing = 5,
                                       brushShape = "box")
```



(a) Large window with  $w_x = w_y = 15$       (b) Small window with  $w_x = w_y = 3$

**Figure 3: A comparison of window sizes for the maximum contrast projection on a segment of the *cells* data set**

The small window size ( $w_x = w_y = 3$ ) shows some artefacts, i.e. several cells seem to be incorrectly projected from different z-stacks. The large window size ( $w_x = w_y = 15$ ) ensures that any single cell is projected from the same z-stack.

## MaxContrastProjection

Similarly, the `smoothing` parameter applies a median filter onto the projection index map to avoid harsh jumps between z-stacks. It is unlikely that any given object would rapidly fluctuate between different focal plains and the median filter ensures the smoothness of the projection.

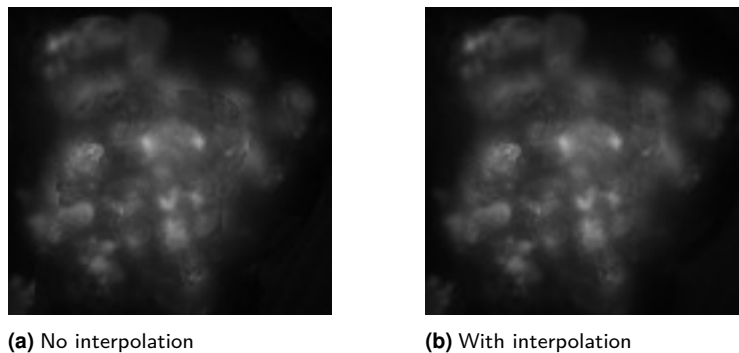
The `brushShape` parameter indicates the shape of the window to be used when determining the contrast at each pixel. By default ("box"), this is a rectangle with the side lengths  $2 \cdot w_x + 1$  and  $2 \cdot w_y + 1$ . Currently supported is also a circle ("disc"). Depending on the geometry of the structures of interest in the image, different window shapes may have an impact on the quality of the projection. For radially symmetrical shapes, such as "disc", the parameter  $w_y$  is assumed to be equal to  $w_x$  regardless of whether it was explicitly defined.

It is also possible to retrieve the actual contrast values as well as the projection index map. The contrast stack is simply the contrast, i.e. variance, for every single pixel of every image in the input image stack. The projection index map then effectively indicates the z-layer with the maximum contrast for every pixel in the  $x - y$ -plane (after optional smoothing).

```
contrastStack = getContrastStack(imageStack = cells, w_x = 15, w_y = 15,
                                brushShape = "box")
indexMap = getIndexMap(contrastStack = contrastStack, smoothing = 5)
max_contrast_fromMap = projection_fromMap(imageStack = cells,
                                          indexMap = indexMap)
```

The maximum contrast projection can then be executed with just the image stack as well as the index map using `projection_fromMap`. In fact, any projection index map can be used here so long as all its values lie between 1 and the number of z-stacks in the input image stack.

A problem of the contrast projection is that if an object lies in multiple focal planes, the contrast projection generates artificial boundaries between these individual areas. This is an unavoidable consequence of imaging an object in discrete layers. Fig. 4 shows what this looks like. In principle, the projection index map is blurred, so that boundaries between focal regions are softened. The projection is then a weighted linear combination of two values. For example, if the projection index map at a given pixel  $(x, y)$  has a value of 7.4, then the value of that pixel in the projection is  $I_{xy} = 0.6 \cdot I_{xy}^{(7)} + 0.4 \cdot I_{xy}^{(8)}$ , where  $I_{xy}^{(j)}$  is the intensity of the pixel in the  $j$ -th layer of image stack.



(a) No interpolation

(b) With interpolation

**Figure 4: A comparison of the effect of linear interpolation on the maximum contrast projection**

Note that the central area in Fig. 4a is distinctly separated from the outside of the organoid. This separation is undesirable as it may lead to issues with image segmentation later on.

## MaxContrastProjection

Lastly, this package also includes the possibility for other projection types. Currently, these are the maximum and minimum intensity projection, the mean and median intensity projection, the standard deviation projection, which calculates the standard deviation of the z-stack at each pixel in the  $x - y$ -plane, and the sum projection, which simply adds together all the values of the z-stack at each pixel in the  $x - y$ -plane.

```
max_intensity_proj = intensityProjection(imageStack = cells, projType = "max")
min_intensity_proj = intensityProjection(imageStack = cells, projType = "min")
mean_intensity_proj = intensityProjection(imageStack = cells,
                                         projType = "mean")
median_intensity_proj = intensityProjection(imageStack = cells,
                                           projType = "median")
sd_intensity_proj = intensityProjection(imageStack = cells, projType = "sd")
sum_intensity_proj = intensityProjection(imageStack = cells, projType = "sum")
```

## 5 Session Info

---

```
toLatex(sessionInfo())
```

- R version 3.5.1 Patched (2018-07-12 r74967), x86\_64-pc-linux-gnu
- Locale: LC\_CTYPE=en\_US.UTF-8, LC\_NUMERIC=C, LC\_TIME=en\_US.UTF-8, LC\_COLLATE=C, LC\_MONETARY=en\_US.UTF-8, LC\_MESSAGES=en\_US.UTF-8, LC\_PAPER=en\_US.UTF-8, LC\_NAME=C, LC\_ADDRESS=C, LC\_TELEPHONE=C, LC\_MEASUREMENT=en\_US.UTF-8, LC\_IDENTIFICATION=C
- Running under: Ubuntu 16.04.5 LTS
- Matrix products: default
- BLAS: /home/biocbuild/bbs-3.8-bioc/R/lib/libRblas.so
- LAPACK: /home/biocbuild/bbs-3.8-bioc/R/lib/libRlapack.so
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils
- Other packages: EBLImage 4.24.0, MaxContrastProjection 1.6.0
- Loaded via a namespace (and not attached): BiocGenerics 0.28.0, BiocManager 1.30.3, BiocStyle 2.10.0, RCurl 1.95-4.11, Rcpp 0.12.19, abind 1.4-5, backports 1.1.2, bitops 1.0-6, compiler 3.5.1, digest 0.6.18, evaluate 0.12, fftwtools 0.9-8, grid 3.5.1, highr 0.7, htmltools 0.3.6, htmlwidgets 1.3, jpeg 0.1-8, knitr 1.20, lattice 0.20-35, locfit 1.5-9.1, magrittr 1.5, parallel 3.5.1, png 0.1-7, rmarkdown 1.10, rprojroot 1.3-2, stringi 1.2.4, stringr 1.3.1, tiff 0.1-5, tools 3.5.1, yaml 2.2.0