Author Information

Christian "Tischi" Tischer

e-mails: christian.tischer@embl.de, tischitischer@gmail.com

Recommended literature on bioimage analysis

- http://www.imaging-git.com/olympus-website-bioimage-data-analysis

...

Inspection of the numerical content of images

An image essentially is an array of numbers with some metadata. For scientific image analysis it is very important to constantly inspect the numeric content of our images, for instance to check whether the image was acquired properly, or whether a mathematical operation such as background subtraction had the desired effect.

Activity: Image inspection

Let's open an image and explore different tools to inspect the numbers in this image. We start by inspection an 8-bit image, where the numbers range from 0 to 255 (2^8-1); we'll explore different bit depths later.

- Open image "../image-inspection/B.tif" [File > Open]

Mouse over

Simply move with the mouse over the image; the intensity will be shown in ImageJ's menu bar.

Pixel inspection tool

- Menu bar: [Px]

Intensity line profile

- Menu bar: Select the line profile tool
- [Analyze > Plot Profile]

Histogram

- [Analyze > Histogram]

Lookup tables (LUTs)

LUTs assign a certain color to each numerical value. Intensity differences are best seen using a grayscale LUT. Choosing the LUT color similar to the emission color of the imaged fluorophore can also make sense. LUTs with multiple colors (e.g., "Fire" in ImageJ) are good for simultaneously seening very dim and very bright images. Finally, LUTs where only the lowest and highest value have a certain color are useful for microscopy, e.g. to indicate saturated pixels.

Activity: Adjust Brightness & Contrast

While the colors in a given LUT are fixed, one can change how these colors are mapped onto the numbers in the image.

- Open image "../image-inspection/B.tif" [File > Open]
- Change the LUT settings:
  - [Image > Adjust > Brightness/Contrast]

Note that this does not change the numbers but only the appearance on your screen.

Important: Don't press [Apply] as this will in fact change the pixel values.

Activity: Explore different LUTs

- Open image "../image-inspection/B.tif" [File > Open]
- Explore different LUTs [Image > Lookup Tables], e.g.
  - Grays
  - HiLo
    - Red: highest, Blue: lowest
    - Important note: "highest" and "lowest" depend on your Brightness&Contrast settings!
  - Fire

Important numerical properties of microscopy images

Background (offset)

Dynamic range

Saturation

Practical activity: Image content inspection

In this activity we will open several images and find out which "issues" they have. The aim is to assign each image to one of the following issues:

- no problem
- high background
- too low background
- low dynamic range
- too much saturation

Use below workflow to inspect the images:

- Open “../image-inspection/A.tif” [File > Open]
- Also open B.tif, C.tif, D.tif, E.tif
- Use below methods to inspect the images and find their "issues"
  - Adjust the display [Image > Adjust > Brightness/Contrast]
  - Examine gray values in whole image [Analyze > Histogram]
  - Analyze gray values along a line [Analyze > Plot Profile]

Additional tasks:

- Lets find five or more different ways to identify saturated pixels in an image

Image bit depths

Images can have different bit depths. Let's start by exploring some of the limitations of the 8-bit image that we were dealing with until now.

Activity: Exploring the limitations of an 8-bit image

- Open image "../image-inspection/B.tif" [File > Open]
- Adding numbers:
  - Copy the original image [Image > Duplicate]
  - Add 500 to each pixel in the image [Process > Math > Add]
  - Inspect the gray values!
- Subtracting numbers:
  - Copy the original image [Image > Duplicate]
  - Subtract 100 from each pixel in the image [Process > Math > Subtract]
  - Inspect the gray values!
- Dividing numbers:
  - Copy the image [Image > Duplicate]
  - Divide each pixel in the image by 2 [Process > Math > Divide]
  - Inspect the gray values!

Obviously this is not what we want since it is all wrong :-).

Activity: Exploring properties of floating point images

- Open image "../image-inspection/B.tif" [File > Open]
- Duplicate the image and already name "32-bit" [Image > Duplicate]
- Convert to 32-bit floating point [Image > Type > 32 bit]
- Inspect the gray values! Did they change after the conversion to 32 bit?
- Now let's repeat above activity

Much better, right?!

Image bit depths in ImageJ

- 8-bit
  - integers from 0-255 (2^8-1)
- 16-bit
  - integers from 0-65535 (2^16-1)
- 32-bit floating point
  - can have negative numbers, such as -1
  - can have non-integer numbers, such as 1.5 or -3.2
  - this format is generally recommended as soon as you do any kind of mathematical operations on your images
  - disadvantage: needs more memory and disk space

Although ImageJ does not support it, your images could also have been acquired with cameras of different bit depth such as 12 or 14 bit.

Image bit depth conversions

Image bit depth conversion is something that you should generally avoid, but sometimes you can't; either because you need to save disk space or because certain operations or plugins only work with certain bit depths. Let's thus explore now what happens if you do convert between different bit depths.

Activity: Conversion from 8-bit to 32-bit floating point

- Open image "../image-inspection/B.tif" [File > Open]
- Duplicate the image and rename it to "32-bit" [Image > Duplicate]
- Convert to 32-bit floating point [Image > Type > 32 bit]
- Inspect the gray values! Did they change after the conversion?

Activity: Conversion from 16-bit to 32-bit floating point

- Open image "../image-format-conversion/16bit.tif" [File > Open]
- Duplicate the image and rename it to "32-bit" [Image > Duplicate]
- Convert to 32-bit floating point [Image > Type > 32 bit]
- Inspect the gray values! Did they change after the conversion?

Activity: 16-bit to 8-bit conversion

OK! Now comes the **tricky part**, where several projects were going very wrong in the past!

- Open "../image-format-conversion/16bit.tif" [File > Open]
- Inspect the gray values: What are the minimum and maximum? Note them down.
- Adjust the display such that it looks nice [Image > Adjust > Brightness/Contrast]
- Convert to 8-bit [Image > Type > 8bit]
- Inspect the gray values again: What are the minimum and maximum now?

Hopefully you are **shocked** that we all got different results! How can this be?

Discussion: How to convert 16-bit to 8-bit

- 0, 65535 => 0, 255
  - Preserves intensities but looses dynamic range
- min, max => 0, 255
  - Maximizes dynamic range, but looses intensity information
- minLUT, maxLUT => 0, 255
  - Leave it up to the user!
  - This is what ImageJ is doing!

Image format conversion

Unfortunately there are many different image formats and since not all software can open all formats you most likely will have to sometimes save your images in different formats. It is of utmost importance that you check what happens to the numerical content of your images when you are doing this! So let's practice!

Activity: Save an image in different formats and inspect how this affects its numerical content and file size

- Open “../image-format-conversion/16bit.tif” [File > Open]
  - Adjust the display such that you actually see something [Image > Adjust > Brightness/Contrast]
- Save as **Jpeg** using different levels of compression (quality)
  - Adjust Jpeg quality (0-100) to 10 [Edit > Option > Input/Output]
  - save as "quality_10.jpg" [File > Save As > Jpeg]
  - repeat for Jpeg qualities 75, and 100
- Save as **png** [File > Save As > PNG]
- Adjust the display such that the image **appears saturated** [Image > Adjust > Brightness/Contrast]
  - Save as "saturated.jpg" [File > Save As > Jpeg]
  - Save as "saturated.png" [File > Save As > PNG]

Now lets go to the folder where you saved the images and check their file size! And, even more important, lets reopen them and check what happened to their gray values!

Image intensity measurements

Mean intensity and sum intensity

=> PowerPoint presentation.

The biophysical meaning of intensities in fluorescence microscopy images

=> PowerPoint presentation.

Practical activity: Manual intensity measurements

- Open “../bit-conversion/16bit.tif”
- Draw a region around a nucleus, e.g. using ImageJ's Polygon Selection
  - Since we will do sum intensity measurements with proper background subtraction you should draw this region rather generously not to miss any intensities!

- Add region to ROI manager [Analyze > Tools > ROI Manager > Add)
- Name the region “nucleus_1” [Analyze > Tools > ROI Manager > Rename]
- Repeat above steps for a background ROI and another nucleus
- Select measurements [Analyze > Set Measurements]:
  - [x] Mean gray value
  - [x] Area
  - [x] Integrated density
    - This measures the sum intensity; in fact it will output two values; the "good" one is the **RawIntDen**, which really simply adds up the gray values in the measurement ROI.
- Select all regions and measure [ROI Manager > Measure]

Now we need to do the proper background subtraction for the two nuclei ROIs, using below formula:

```
Sum_BgCorr = RawIntDen - Area * Mean_Background
```

In words, we subtract for each pixel in the ROI the mean value of the background.

Intensity measurements and their interpretation, with local background

Intensity measurements are a **very tricky business**, not because they are technically difficult, but because one can make many mistakes in the interpretation of the numbers. This very easily leads to wrong scientific conclusions!

Activity: Intensity measurements with local background subtraction

Let's first open the images:

- Open all images in this folder "../dna-damage-synthetic-data/"

We pretent that these are **widefield microscopy** images of one nucleus where a GFP-tagged DNA damage repair enzyme is diffusing around. In some of the images a well controlled laser cut was induced and thus the DNA repair enzyme binds the damage site. Some images say "Treated" in their title. The idea is that the scientist added a drug and wanted to find out if this drug enhances or diminishes the binding of the DNA repair enzyme.

Examine with line profile

Let's first look at the images using an intensity line profile and discuss what we see.

Measure fraction of protein bound to damage site

Ok, now let's try to really measure a bio-physically meaningful number. This is generally challenging and one really has to think about it!

In this case, assuming

- this is widefield microscopy, and
- the unbound molecule is fast diffusing
- the laser cut had the exact same effect in all experiments

..it probably makes sense to divide the sum intensity of the bound protein by the sum intensity in the nucleus; i.e. total_bound / total_available.

To do this we need to measure:

- Mean intensity outside the nucleus (mean_bg)
- Mean intensity next to damage site inside the nucleus (mean_nucleus_diffusive)
- Sum intensity of nucleus and area of corresponding ROI (sum_nucleus, area_nucleus_ROI)
- Sum intensity of damage site and area of corresponding ROI (sum_damage, area_damage_ROI)

Now we need to compute:

- total_signal_nucleus = sum_nucleus - area_nucleus_ROI * mean_bg
- total_signal_damage = sum_damage - area_damage_ROI * mean_nucleus_diffusive

And finally:

- fraction_bound_to_damage = total_signal_damage / total_signal_nucleus

Hard work, right? And many options to make little mistakes, thus we only should preform intensity measurements with utmost care!

Discussion points

- How do our observations relate to this: https://en.wikipedia.org/wiki/Binding_constant
- Divide by the length (and or width) of the laser damage cut?
    - length makes sense
    - width probably not
- What about computing the mean intensity in the nucleus next to the damage site?
    - In principle attractive, because for a KD we need the concentration of the unbound protein.
    - For a confocal image this can make sense if there are not many substructures in the nucleus (like nucleoli). Basically, if the concentration of the protein is homgeneous in the region where we measure the mean (you have to think in 3-D; PSF!) this mean intensity gives information about the concentration of unbound protein; however as soon as there is a lot of structure in the signal it is not clear that it helps
    - For a widefield image it is kind of the same argument, however the 3D shape of the measured region is infinitely big! One can see that one measures a larger amount of protein in the center than at the edge of the nucleus.
- What changes if we assume that this is **one confocal slice** rather than a widefield image?

Image segmentation

https://en.wikipedia.org/wiki/Image_segmentation says: In computer vision, image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as super-pixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics.

Applications of image segmentation in biology

- object counting
- object localization measurements
- object shape measurements
- object intensity measurements
- in general: object **feature** measurements

In general, image segmentation typically is a two step process, where you

1. identify all pixels that potentially belong to an object
2. group pixels together belonging to one object (as you typically have several objects in one image).

Activity: Manual global thresholding followed by "particle analysis"

In fluorescence microscopy, image segmentation often is easy, because the objects of interest are simply brighter than the "background".

Let's try:

- Configure image segmentation settings [Process > Binary > Options]:
  - ☑ Black Background
- Open image: "../image-inspection/B.tif" [File > Open]
- Manually adjust a threshold value [Image > Adjust > Threshold]
  - You may press [Apply] but you do not have to; it also works with the "red" overlay.
- Perform a "connected component analysis" [Analyze > Analyze Particles]
  - Other wordings are: "object detection", "particle analysis"
- Run it again and explore the different options of the "Particle Analyzer"

Discussion

- Single threshold vs. 'gating'
- Object size and shape filtering
- Different types of object representations (pros and cons)

The signal to noise (S/N) ratio

At the microscope, especially setting up a live cell experiment where you want to avoid photo-bleaching of your fluorophore, you typically wonder: "How good does the image need to be in order for me to be able to still segment the objects?"

A very important concept in this regard is the signal to noise ratio (S/N), which, in my humble opinion, is often confused with the much less important signal to background ratio (S/B).

Let's have a look and try to segment nuclei of different intensities:

- Open "../signal-to-noise/hb2-mCherry.tif" [File > Open]
- Now try to threshold the nuclei [Image > Adjust > Threshold]
  - You see that this is easy for the bright ones but does not really work for the very dim ones (you may have to adjust the LUT settings [Image > Adjust > Brightness/Contrast] to even see the dark ones).

Let's now try to quantify why it is difficult to segment the dark nuclei by measuring their S/N.

- [Analyze > Set Measurements]:
  - ☑ Mean gray value
  - ☑ Standard deviation
- Draw an ROI inside a dim nuclues of your choice, e.g. using the “Oval Selection”
- Save that ROI [Analyze > Tools > ROI Manager > Add]
  - ...and give it a good name [Analyze > Tools > ROI Manager > Rename]
- Now also draw and save an ROI in the background right next to the nucleus
- Select both regions and measure them [ROI Manager > Measure]

Now let's apply below formulas to measure the S/N and S/B:

- S/N = (Mean_Nucleus - Mean_Background) / Sdev_Background

- S/B = Mean_Nucleus / Mean_Background

Discussion

As mentioned, although sometimes used, I don't understand the use of S/B. For S/N however it is very clear that if you are getting as low as two, you start getting into trouble in terms of being able to still segment this object.

Image filtering (convolution)

Image filtering is a very wide field, where mostly one replaces the intensity of each pixel by some mathematical function of its neighbors. The most simple example is probably the 3x3 mean filter, where each pixel is replaced by the mean value in a 3x3 neighborhood (i.e. inclucing the pixel itself and its 8 neighbors).

Activity: Manually compute a 3x3 mean and a 3x3 median filter

Mean and median can be almost the same, but, depening on the data, also be very different; thus let's compare them. In fact, the median filter is quite important for local background subtraction, while the mean filter is less useful in this context (see below).

Let's compute a 3x3 mean and a 3x3 median filter for the central pixel in below examples and compare the results.

10	11	10	13	12
13	1000	10	11	14
21	15	**11**	13	10
14	13	12	11	10
11	11	10	13	12

What do you get for mean and median?

10	11	10	1	1
13	12	10	1	1
21	15	**11**	0	0
14	13	12	0	0
11	11	10	1	0

And what do you get here for mean and median?

Discussion

From above examples, it should have become clear why a median filter is called both:

- robust to outliers
- edge preserving

Activity: Segmentation of noisy images with the help of filtering (smoothing)

- Open: "../signal-to-noise/noisy-nuclei.tif" [File > Open]

- Try to threshold the image [Image > Adjust > Threshold]
  - This doesn't really work, right?
- Let's try to smooth the image first, e.g. using
  - A 11x11 median filter [Process > Filters > Median]
    - Radius: 5 (2*5 + 1 = 11)
- Now, let's segment the nuclei, using:
  - [Image > Adjust > Threshold]
  - [Analyze > Analyze Particles]

Image convolution

The 3x3 mean filter that we applied above can also be expressed in terms on a "convolution", with a "convolution kernel" as show below:

Kernel:

1/9	1/9	1/9
1/9	**1/9**	1/9
1/9	1/9	1/9

Original image:

15	11	5
14	**13**	12
11	11	10

Intermediate step:

15 * 1/9	11 * 1/9	5 * 1/9
14 * 1/9	**13** * 1/9	12 * 1/9
11 * 1/9	11 * 1/9	10 * 1/9

Sum is 11.33333

Convolved image:

?	?	?
?	**11.333**	?
?	?	?

Basically, you multiply each pixel in the original image with the number that is written in the kernel and then you replace the center pixel with the sum of all pixels.

Activity: Try the effect of different convolution kernels

- Open any image of your choice
- [Process > Filters > Convolve]

Intensity measurements with automated local background subtraction

Above we already practiced how to subtract a local background manually; let's try to automated this. This is important for image batch analysis or when the local background is uneven and not easy to subtract manually.

Activity: Automated local background subtraction

Example data:

- We need to run this macro to generate example data:
  - "../dna-damage-synthetic-data/make-images--dna-damage-synthetic-data.py"
  - [Help > Update > Manage Update Sites] ImageScience
  - [Run] to generate the images

We pretent that these are **widefield microscopy** images of one nucleus where a GFP-tagged DNA damage repair enzyme is diffusing around. In some of the images a well controlled laser cut was induced and thus the DNA repair enzyme binds the damage site. Some images say "Treated" in their title. The idea is that the scientist added a drug and wanted to find out if this drug enhances or diminishes the binding of the DNA repair enzyme.

Workflow:

- Background subtraction
- Separate intensity in damage site (bound protein) from unbound protein
  - Local background subtraction alternatives:
    - Manual
      - on image
      - in R/Excel
    - Median subtraction (32bit!)
    - Tophat filter
- Measure sum intensity in damage site (including bg-subtraction)
- Measure sum intensity in nucleus

Discussion:

- How do our observations relate to this: https://en.wikipedia.org/wiki/Binding_constant
- Divide by the length (and or width) of the laser damage cut?
  - length makes sense
  - width probably not
- What about computing the mean intensity in the nucleus next to the damage site?
  - In principle attractive, because for a KD we need the concentration of the unbound protein.
  - For a confocal image this can make sense if there are not many substructures in the nucleus (like nucleoli). Basically, if the concentration of the protein is homgeneous in the region where we measure the mean (you have to think in 3-D; PSF!) this mean intensity gives information about the concentration of unbound protein; however as soon as there is a lot of structure in the signal it is not clear that it helps
  - For a widefield image it is kind of the same argument, however the 3D shape of the measured region is infinitely big! One can see that one measures a larger amount of protein in the center than at the edge of the nucleus.
- Background subtraction must be done with floating point
  - Not for tophat though