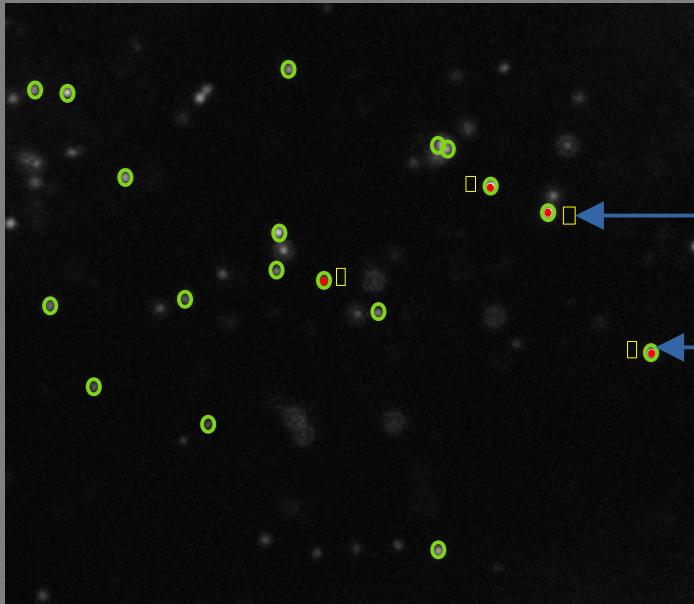


« Single mRNA » calibration

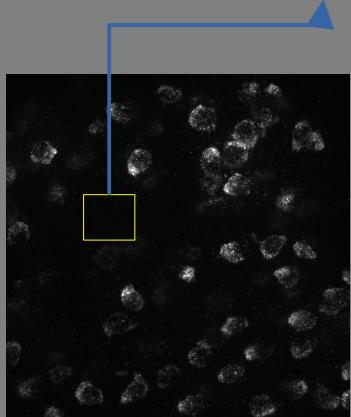


Xml file to describe dots positions

Roi file to describe dots rois background

Dots segmentation

$$mean\ dot\ Int.\ bg = \frac{\sum_{dot\ zmin}^{dot\ zmax} roi\ Integ.\ Int.}{\sum_{dot\ zmin}^{dot\ zmax} roi\ area}$$



$$Corrected\ single\ mRNA\ Integrated\ Int. = \frac{\sum_1^n dot\ Integ.\ Int. - \sum_1^n mean\ dot\ bg\ Int.* \sum_1^n dot\ vol}{n}$$

mRNA Scope parameters

Parameters

Channels

DAPI : CSU-405
Gene Reference : CSU-561
Gene X : CSU-642

nucleus parameters

Threshold method : Otsu
Min Volume size : 50.00
Max Volume size : 900.00
Nucleus dilatation : 3.00
Section to remove : 0

Single dot calibration

Gene reference single dot intensity : 0
Gene X single dot intensity : 0

Background detection

Auto background : From roi
Size of background box size : 100

Gene reference background intensity from calibration : 100
Gene X background intensity from calibration : 100

Remove ghost dots

Cancel OK

Channels selection

Threshold method for nucleus segmentation

Nucleus parameters

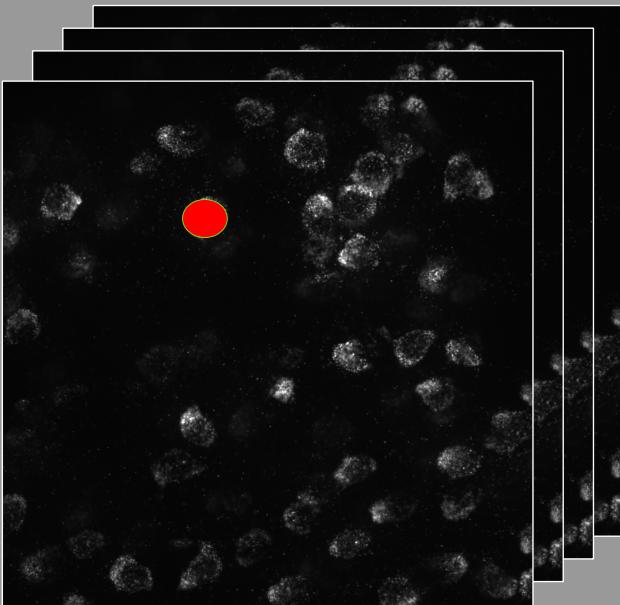
Single mRNA intensity parameters obtained from calibration

Background detection methods

Background parameters

Background from calibration

Take the background value computed from calibration process



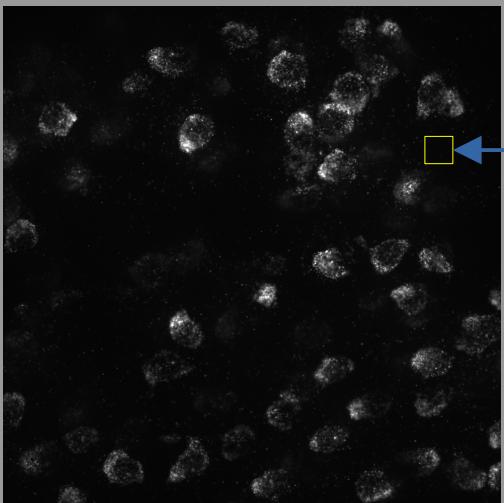
Gene X stack

$$\text{mean cell bg Int.} = \frac{\sum_1^n \text{meandot bg Int.} * \sum_1^n \text{dot vol}}{n}$$

$$\text{Corrected cell integrated intensity} = \text{cell integrated intensity} - (\text{mean cell bg Int.} * \text{cell vol})$$

Background from manual roi

Stack gene image



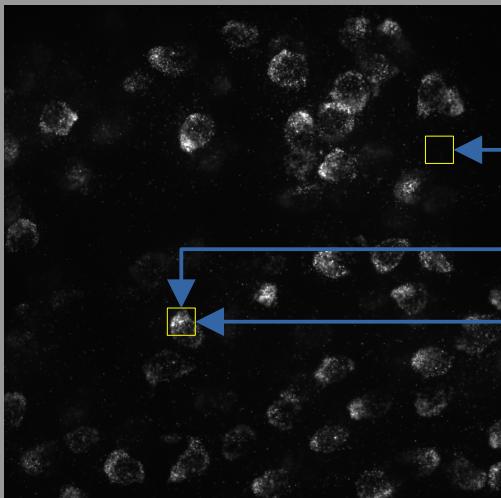
$$mean\ cell\ Int.\ bg = \frac{\sum_{zmin}^{zmax} roi\ Integ.\ Int.}{\sum_{zmin}^{zmax} roi\ area}$$

```
/*
 * Get Mean of intensity in stack
 */
public static double find_background(ImagePlus img, int zMin, int
zMax) {
    ResultsTable rt = new ResultsTable();
    Analyzer ana = new Analyzer(img,
Measurements.INTEGRATED_DENSITY, rt);
    double intDen = 0;
    int index = 0;
    for (int z = zMin; z <= zMax; z++) {
        img.setSlice(z);
        ana.measure();
        intDen += rt.getValue("RawIntDen", index);
        index++;
    }
    double vol = img.getWidth() * img.getHeight() * (zMax - zMin + 1);
    double bgInt = intDen / vol;
    rt.reset();
    return(bgInt);
}
```

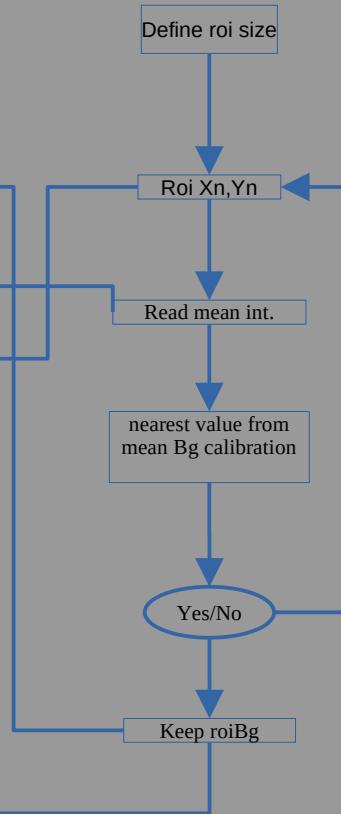
Auto background detection

Find roi with nearest value computed from calibration process

Stack gene image



$$mean\ cell\ Int.\ bg = \frac{\sum_{zmin}^{zmax} roi\ area}{\sum_{zmin}^{zmax} roi\ Integ.\ Int.}$$



```

/**
 * Find nearest background roi
 * @param img
 * @param size
 * @return
 */
public static Roi findRoiBbackgroundAuto(ImagePlus img, double bgGene) {
    // scroll gene image and measure bg intensity in roi
    // take roi at intensity nearest from bgGene

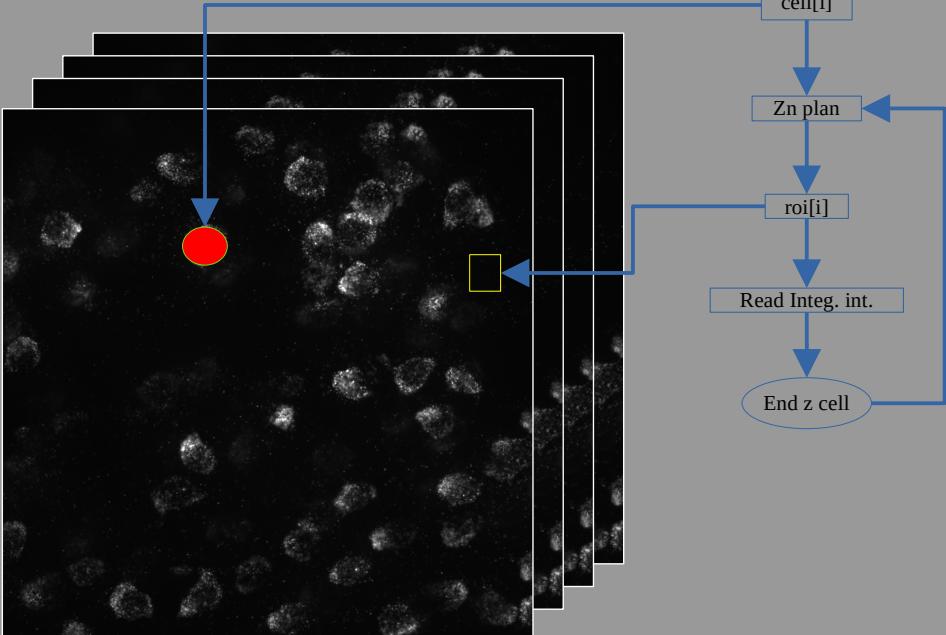
    ArrayList<RoiBg> intBgFound = new ArrayList<RoiBg>();

    for (int x = 0; x < img.getWidth() - roiBgSize; x += roiBgSize) {
        for (int y = 0; y < img.getHeight() - roiBgSize; y += roiBgSize) {
            Roi roi = new Roi(x, y, roiBgSize, roiBgSize);
            img.setRoi(roi);
            ImagePlus imgCrop = img.crop("stack");
            double bg = find_background(imgCrop, 1, img.getNSlices());
            intBgFound.add(new RoiBg(roi, bg));
            closeImages(imgCrop);
        }
    }
    img.deleteRoi();
    // sort RoiBg on bg value
    intBgFound.sort(Comparator.comparing(RoiBg::getBgInt));

    // Find nearest value from bgGene
    double min = Double.MAX_VALUE;
    double closest = bgGene;
    Roi roiBg = null;
    for (RoiBg v : intBgFound) {
        final double diff = Math.abs(v.getBgInt() - bgGene);
        if (diff < min) {
            min = diff;
            closest = v.getBgInt();
            roiBg = v.getRoi();
        }
    }
    int roiCenterX = roiBg.getBounds().x+(roiBgSize/2);
    int roiCenterY = roiBg.getBounds().y+(roiBgSize/2);
    System.out.println("Roi auto background found = "+closest+" center x = "+roiCenterX+", y = "+roiCenterY);
    return(roiBg);
}
  
```

Background detection

Cell background

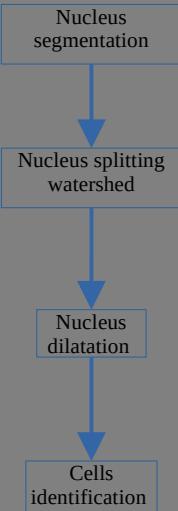
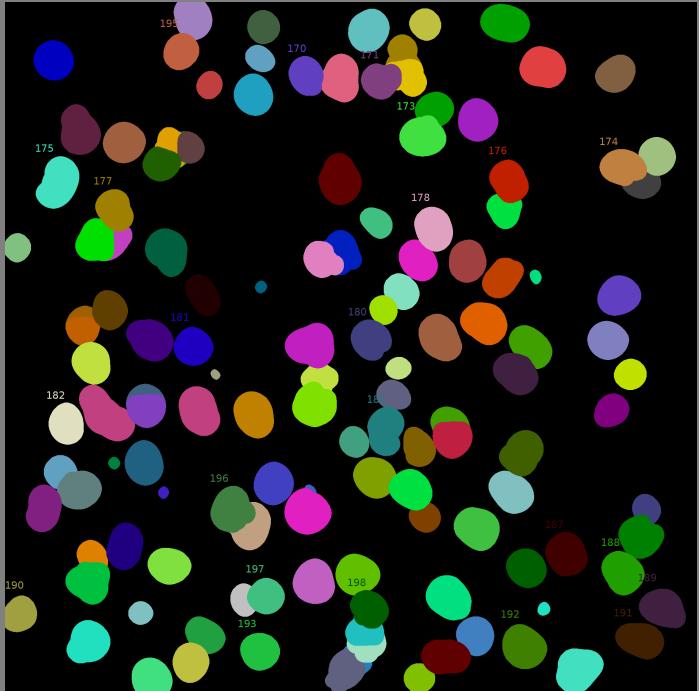


Gene X stack

```
/*
 * Get Mean of intensity in roi cropped stack
 */
public static double find_background(ImagePlus img, int zMin, int zMax) {
    ResultsTable rt = new ResultsTable();
    Analyzer ana = new Analyzer(img, Measurements.INTEGRATED_DENSITY, rt);
    double intDen = 0;
    int index = 0;
    for (int z = zMin; z <= zMax; z++) {
        img.setSlice(z);
        ana.measure();
        intDen += rt.getValue("RawIntDen", index);
        index++;
    }
    double vol = img.getWidth() * img.getHeight() * (zMax - zMin + 1);
    double bgInt = intDen / vol;
    rt.reset();
    return(bgInt);
}
```

$$mean\ bg\ Int.\ (cell[i]) = \frac{\sum_{cell\ zmin}^{cell\ zmax} roi\ Integ.\ .Intensity}{\sum_{cell\ zmin}^{cell\ zmax} roi\ area}$$

Cells detection

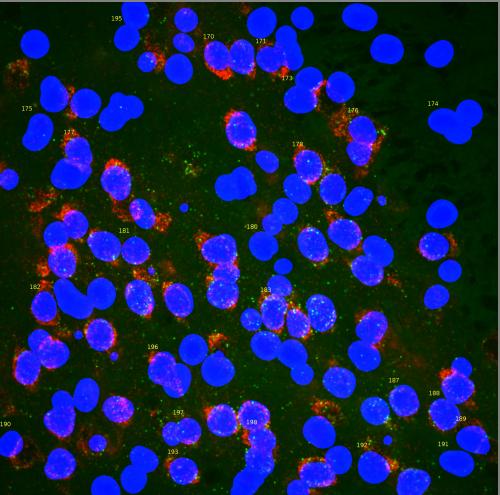


```
/*
 * Nucleus segmentation 2
 * @param imgNuc
 * @return cellPop
 */
public static Objects3DPopulation find_nucleus2(ImagePlus imgNuc) {
    ImagePlus img = new Duplicator().run(imgNuc);
    ImageStack stack = new ImageStack(img.getWidth(), imgNuc.getHeight());
    for (int i = 1; i <= img.getStackSize(); i++) {
        IJ.showStatus("Finding nucleus section "+i+" / "+img.getStackSize());
        img.setZ(i);
        img.updateAndDraw();
        IJ.run(img, "Nuclei Outline", "blur=20 blur2=30 threshold_method=" + threshold + " outlier_radius=50 outlier_threshold=1
max_nucleus_size=100"
                + "min_nucleus_size=10 erosion=5 expansion_inner=5 expansion=5 results_overlay");
        img.setZ(1);
        img.updateAndDraw();
        ImagePlus mask = new ImagePlus("mask", img.createRoiMask().getBufferedImage());
        ImageProcessor ip = mask.getProcessor();
        ip.invertLut();
        for (int n = 0; n < 3; n++) {
            ip.erode();
            stack.addSlice(ip);
        }
        ImagePlus imgStack = new ImagePlus("Nucleus", stack);
        IJ.showStatus("Starting watershed... ");
        ImagePlus imgWater = WatershedSplit(imgStack, 8);
        closeImages(imgStack);
        imgWater.setCalibration(imgNuc.getCalibration());
        Objects3DPopulation cellPop = new Objects3DPopulation(imgWater);
        cellPop.removeObjectsTouchingBorders(imgWater, false);
        closeImages(imgWater);
        closeImages(img);
        return cellPop;
    }
}

public static Objects3DPopulation findNucleus(ImagePlus imgNuc, ImagePlus imgGene) {
    Objects3DPopulation nucPopOrg = new Objects3DPopulation();
    nucPopOrg = find_nucleus2(imgNuc);
    System.out.println("-- Total nucleus Population :" + nucPopOrg.getNbObjects());
    //size filter
    Objects3DPopulation nucPop = new Objects3DPopulation(nucPopOrg.getObjectsWithinVolume(minNucVol, maxNucVol, true));
    int nbNucPop = nucPop.getNbObjects();
    System.out.println("-- Total nucleus Population after size filter: " + nbNucPop);
    //create dilated nucleus population
    Objects3DPopulation cellsPop = new Objects3DPopulation();
    if (nucDil != 0) {
        for (int o = 0; o < nucPop.getNbObjects(); o++) {
            Object3D obj = nucPop.getObject(o);
            //cellsPop.addObject(dilatedObject((float)(nucDil/cal.pixelWidth), (float)(nucDil/cal.pixelHeight), (float)(nucDil)));
            cellsPop.addObject(dilCellObj(imgNuc, obj));
        }
    } else
        cellsPop = findCell(imgGene, nucPop);
    return cellsPop;
}
```

mRNA quantifications

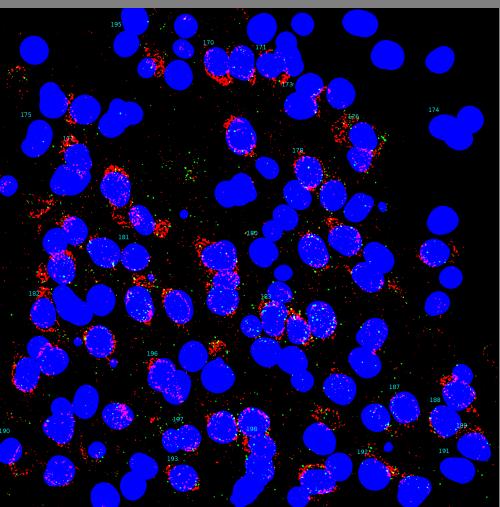
Cell intensity



$$\text{Corrected cell integrated intensity} = \text{cell integrated intensity} - (\text{mean bg Int.} * \text{cell vol})$$

$$\text{Total single mRNA per cell} = \frac{\text{corrected cell Integ. Int.}}{\text{corrected single mRNA Integ. Int.}}$$

Dot intensity

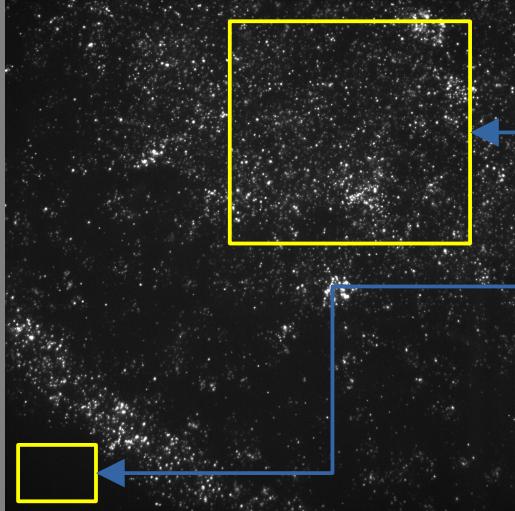


$$\text{Corrected dots integrated intensity} = \text{cell dots integrated intensity} - (\text{mean bg Int.} * \text{cell vol})$$

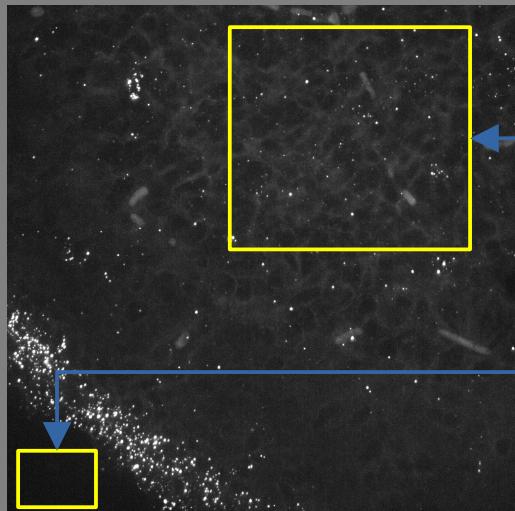
$$\text{Total single mRNA per cell} = \frac{\text{corrected dots cell Integ. Int.}}{\text{corrected single mRNA Integ. Int.}}$$

https://github.com/pmailly/RNA_Scope.git

In the case of compact cells such as in cerebellum



Gene ref. Roi bg



Gene ref. Roi bg

$$mean\ bg\ Int.\ roi = \frac{\sum_{zmin}^{zmax} roi\ Integ.\ .Intensity}{\sum_{zmin}^{zmax} roi\ area}$$

$$gene\ intensity\ density = \frac{roi\ integrated\ intensity - (mean\ bg\ Int.\ *roi\ vol)}{roi\ vol.}$$

https://github.com/pmailly/RNA_Scope.git