**Parameters to adjust:**

filename, path, channels, and pb\_thresh which is the minimum number of nearest-neighbour distances that need to be drawn to a nucleus for it to be considered PB-positive.

A screen shot of a computer

AI-generated content may be incorrect.

**Method:**

Lilli’s survival assay code is used as a framework, but in that code only the overlap between green/red and nuclei are detected. This works well for experiments with just uniform GFP/mCHERRY.

A close-up of a computer screen

AI-generated content may be incorrect.

But for images where PBs are detected, it is a bit more nuanced.

A close-up of a cell survival

AI-generated content may be incorrect.

Here, because the LSM-KD is not 100% efficient, some cells will be counted incorrectly as PBnegative, even though they still have PBs. Thus, we require a way to associate PBs with their respective nuclei and detect that cell as being PBpositive. The method uses a Gaussian Mixture Model to detect PB spots, then draws nearest-neighbour distances between PB spots and DAPI nuclei, as shown. A pb\_thresh is defined, where if a cell has >pb\_thresh associated nearest-neighbour PB spots, it is counted as PBpositive.

A diagram of a cell survival

AI-generated content may be incorrect.

A diagram of a cell

AI-generated content may be incorrect.

**Outputs of the program (saved in Condition/Results folder):**

Annotated figures for each image, showing nuclei that the program deems as PBpositive, PBnegative, or uncertain. The uncertain nuclei have more than 0 nearest-neighbour PBs, but less than the user-specified pb\_thresh.

A csv file tabulating the automatic counting results, for each image (images are named by the DAPI channel filename).

Example file structure:

**A screenshot of a computer

AI-generated content may be incorrect.**

**Example annotated figure:**

A screenshot of a cell phone

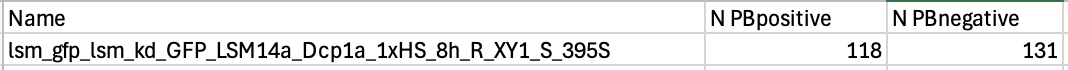
AI-generated content may be incorrect.

After letting the program do the automatic countings, the user should check through the output images and make corrections, especially paying attention to the cream coloured nuclei, which are uncertain but counted by the program as PBnegative. Specifically:

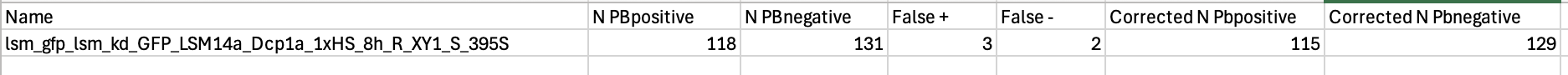
* Disclude weird nuclei (dividing, oddly shaped) – the program will not (I thought this would be easier than entering bad nuclei manually at the start every time)
* Remove false positives

These corrections may be made manually in the final CSV file for every file in the condition.

For example, this would be a default output:



If I check the annotated figure and saw that the program counted 3 false positives, and 2 false negatives, I could do this in Excel:



Some examples of difficult cases to watch out for:

A green and yellow glowing spots

AI-generated content may be incorrect. A close up of a colorful object

AI-generated content may be incorrect.