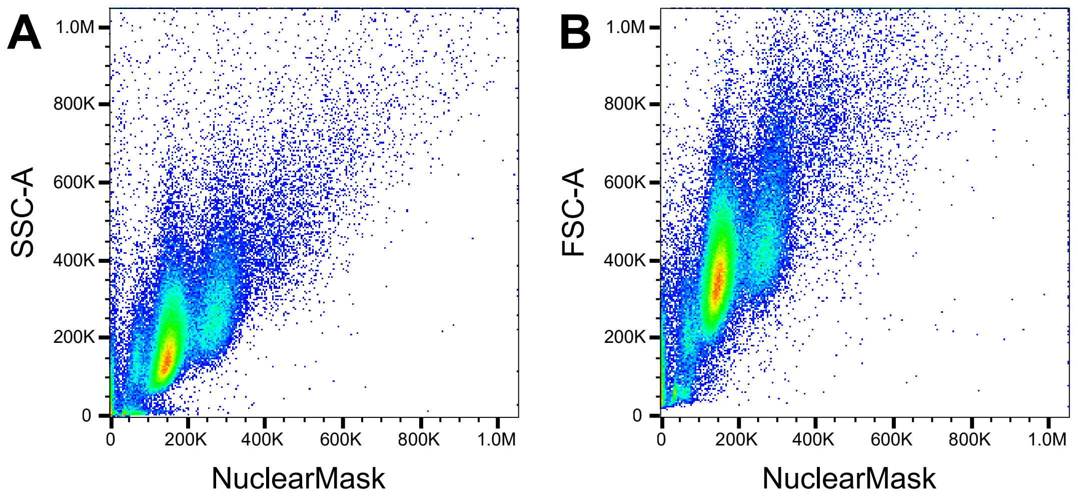

Fig. S1. Flow Cytometry. Cells stained with NuclearMask can be identified by flow cytometry as mononucleated macrophages (fluorescent intensity ≈100K-225K), binucleated macrophages (≈225K-325K), or MGC (>325K). Further binary subclassification of MGC based on number of nuclei may be possible, but a large quantity of cells would be required for accuracy. Dot plots show that side scatter **(A)** and forward scatter **(B)** tend to increase with nuclear fluorescence. However, scatter alone would not be enough to separate MGC from macrophages with high purity. Results concatenated from n=3 true replicates.

SUPPLEMENTARY INFORMATION

Video Narration Script

Time 0:00 = Title Screen
This nuclei counting method is designed to be semi-automated, while giving the user a chance to visually review each result. This video will have three examples to show how the process works, followed by a demonstration of the automated ImageJ macro running in real-time.

Time 0:20 = Sample1
After starting ImageJ, open the image by selecting File>Open. The duplicate command creates a separate window that will be used for counting. This is an image of nuclei that are fluorescently stained. The clear separation between background and nuclei allows for simple image segmentation using the “Make Binary” command, which automatically thresholds based on the image histogram. Occasionally, there may be clumps or partially overlapping nuclei. For more accurate counts, these can be separated using binary operations such as “Watershed” or “Ultimate Points.” “Ultimate Points” tends to be more effective for irregularly-shaped nuclei and simplifies the upcoming particle counting step. The output is then converted to binary again. The “Analyze Particles” command will count each individual point representing a nucleus. The settings do not need to be adjusted for size or circularity because each “Ultimate Point” is just one pixel in size. These settings created a results window and added points to the ROI (Region of Interest) Manager. The ROI manager can be used on the original image to visually check for accuracy of the count. Each number label indicates a nucleus that was counted. If the count was satisfactory, record the results.

Time 1:54 = Sample2
In this next example, the same concepts will be used for analyis. However, the process will be done using macros instead of point-and-click menu options. Each macro line can be run individually by selecting “Macros”>“Evaluate Line” or the shortcut key ctrl+Y. This first line cleared the ROI from previous counts. Running the remaining lines will finish the analysis in the same way as before. Even though this image contained lower quality nuclear staining, this method has the ability to distinguish clumps and some other undesired areas of signal.

Time 2:40 = Sample3
Next is an example of a very low quality image. The full macro will be run at once this time. A qualitative look at the ROI results show that the automated method didn’t work with this low quality image. Therefore, a modified approach should be used in these situations. Similar concepts are used here, except that the thresholding is done manually by the user. Certain images may benefit from adding other processing steps as well, such as using “Fill Holes” in this fuzzy, pixellated example. The remaining steps are the same as before.

Time 3:37 = Real-time demo
Finally, all of these steps have been integrated into a single, more user-friendly macro. This is included with the journal article’s supplementary information. It can be installed as a menu command and then run by using the shortcut key: lowercase q. Now, this demonstrates the complete analysis method in real-time. After the batch of images are complete, counting results in the summary window can be exported.

“CountAndConfirm.ijm” File Script

```
// Macro “CountAndConfirm” by Kevin Trout
// Add or remove "run("Fill Holes");" lines as appropriate for image conditions

macro "CountAndConfirm [q]" {
function PrimaryMethod() {
	roiManager("reset");
	run("Duplicate...", "title=[Counting Window]");
	run("Make Binary");
	// run("Fill Holes");
	run("Ultimate Points");
	run("Make Binary");
	run("Analyze Particles...", "size=0-Infinity circularity=0.00-1.00 show=Nothing
summarize add");
	close();
	roiManager("Show All");
	if(getBoolean("Satisfied with Result?\n  \"Yes\" to continue with next image in
folder.\n  \"No\" to repeat analysis with manual threshold.\n  \"Cancel\" to end batch.")) {
		run("Open Next");
		PrimaryMethod();
	} else {
		n=Table.size("Summary");
		Table.deleteRows(n-1,n-1,"Summary");
		SecondaryMethod();
	}
}

function SecondaryMethod() {
	roiManager("reset");
	run("Duplicate...", "title=[Counting Window]");
	run("8-bit");
	run("Threshold...");
	title = "User Input Required";
	msg = "Manually threshold, then click \"OK\".";
	waitForUser(title, msg);
	run("Make Binary");
	run("Fill Holes");
	run("Ultimate Points");
	run("Make Binary");
	run("Analyze Particles...", "size=0-Infinity circularity=0.00-1.00 show=Nothing
summarize add");
	close();
	roiManager("Show All");
	if(getBoolean("Satisfied with Result?\n  \"Yes\" to continue with next image in
folder.\n  \"No\" to repeat analysis with manual threshold.\n  \"Cancel\" to end batch.")) {
		run("Open Next");
		PrimaryMethod();
	} else {
		n=Table.size("Summary");
		Table.deleteRows(n-1,n-1,"Summary");
		SecondaryMethod();
	}
}
PrimaryMethod();
}
```