Supplementary Information - Simvastatin attenuates endothelial dysfunction in a coronary artery-on-a-chip

Jasneil Singh[1,2,3,*], Deepu Ashok[1,2,3,4,*], Henry Howard[2,4], Steven G. Wise[1,2,3], Anna Waterhouse[1,2,3]

[1]School of Medical Sciences, Faculty of Medicine and Health, The University of Sydney, NSW 2006, Australia

[2]The Charles Perkins Centre, The University of Sydney, NSW 2006, Australia

[3]The University of Sydney Nano Institute, The University of Sydney, NSW 2006, Australia

[4]School of Biomedical Engineering, Faculty of Engineering, The University of Sydney, NSW, 2006, Australia

[*]Authors contributed equally

Note S1 - Python Script – Pump Control

```
1. import numpy as np
 2. import pandas as pd
 3.
 4. # Define pixelPerMicron as a constant
 5. pixelPerMicron = 1.204
 6.
 7. # Read parameters from separate CSV file
 8. parameters_df = pd.read_csv("/Users/jsin6148/Downloads/parameters.csv")
 9.
10. # Loop through each row in the parameters file, skipping the first row
11. for index, row in parameters_df.iloc[22:].iterrows():
12.     filename = row['filename']
13.     bifX = float(row['bifX'])
14.     bifY = float(row['bifY'])
```

```
    bifMean = float(row['bifMean'])

    # Provide the full path to the filename
    full_path_to_file = "/Users/XYZ/Downloads/" + filename

    print("Processing file: {filename}")

    # Read the CSV file into a NumPy array
    data = np.loadtxt(full_path_to_file)

    # Specify parameters
    boxSize = 200
    analyseAreaSize = round(boxSize * pixelPerMicron)
    minDistance = 1140
    xPosition = 0
    yPosition = 0

    # Iterate over the data using array slicing
    for r in range(0, data.shape[0] - (analyseAreaSize - 1), 10):
        for c in range(0, data.shape[1] - (analyseAreaSize - 1), 10):
            # Extract the 2x2 grid
            analyseArea = data[r:r + analyseAreaSize, c:c + analyseAreaSize]

            # Calculate mean intensity
            meanIntensity = np.mean(analyseArea)

            # Check thresholds
            if meanIntensity > bifMean * 2:  # Assuming fold-change threshold is 2
                # Calculate distance
                xDistance = bifX - c
                yDistance = bifY - r
                hypotenuse = (np.sqrt(xDistance ** 2 + yDistance ** 2)) / pixelPerMicron

                if hypotenuse < minDistance and xDistance != 0 and yDistance != 0:
                    minDistance = hypotenuse
```

```
                    xPosition = c
                    yPosition = r

    # Print results for each file
    print("Final min distance:", minDistance, "microns")
    print("This occurs at x:", xPosition, "and y:", yPosition)

print("All files processed.")

```

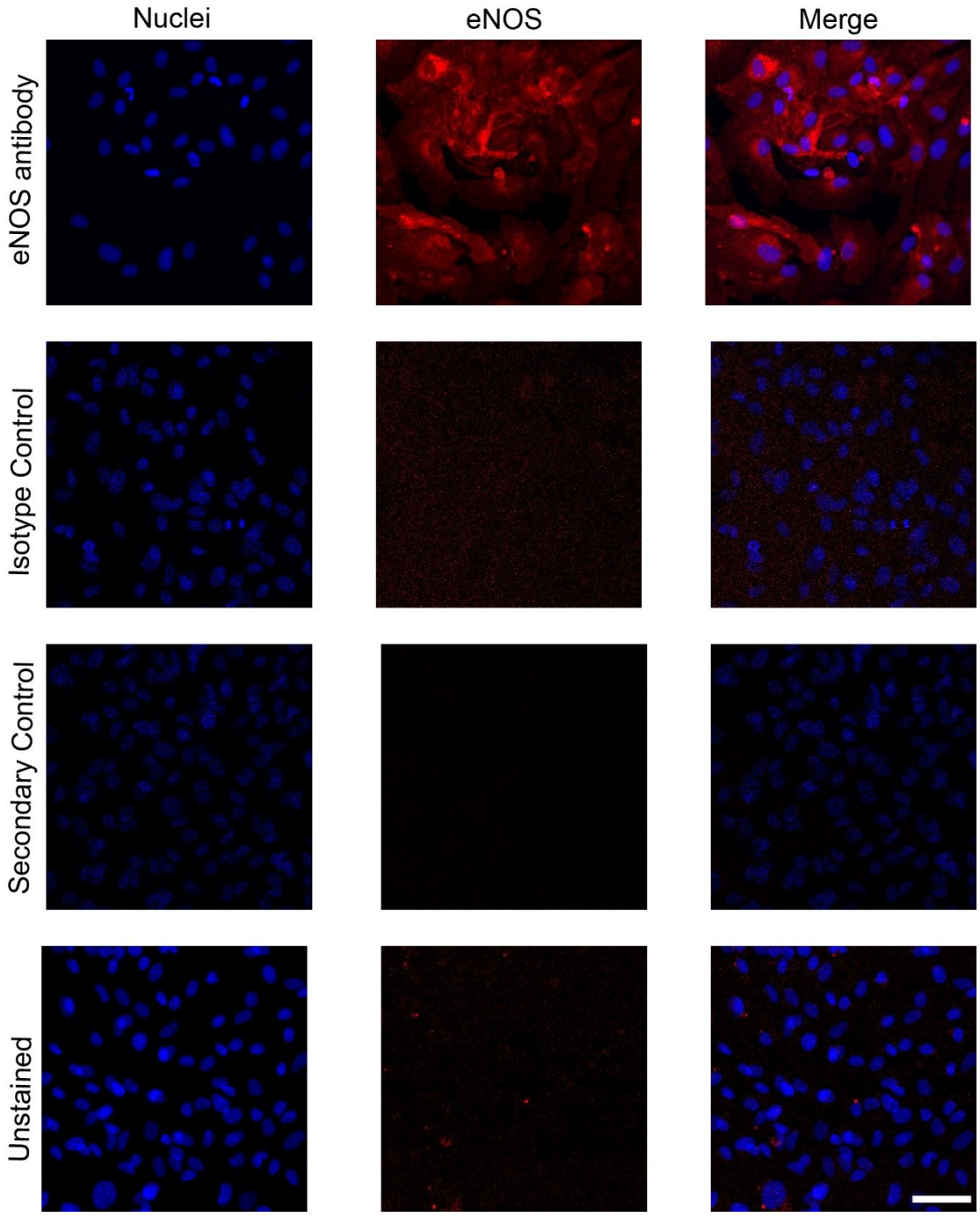

Figure S1- Isotype validation of eNOS immunofluorescence in HCAECs. Representative confocal micrographs of HCAECs labelled with eNOS rabbit anti-human polyclonal antibody showing eNOS signals, rabbit IgG isotype control showing low background signals, secondary goat anti-rabbit IgG alone showing low background signal, and the unstained group showing minimal fluorescence. Nuclei were stained with Hoescht. The scale bar is 100 µm.

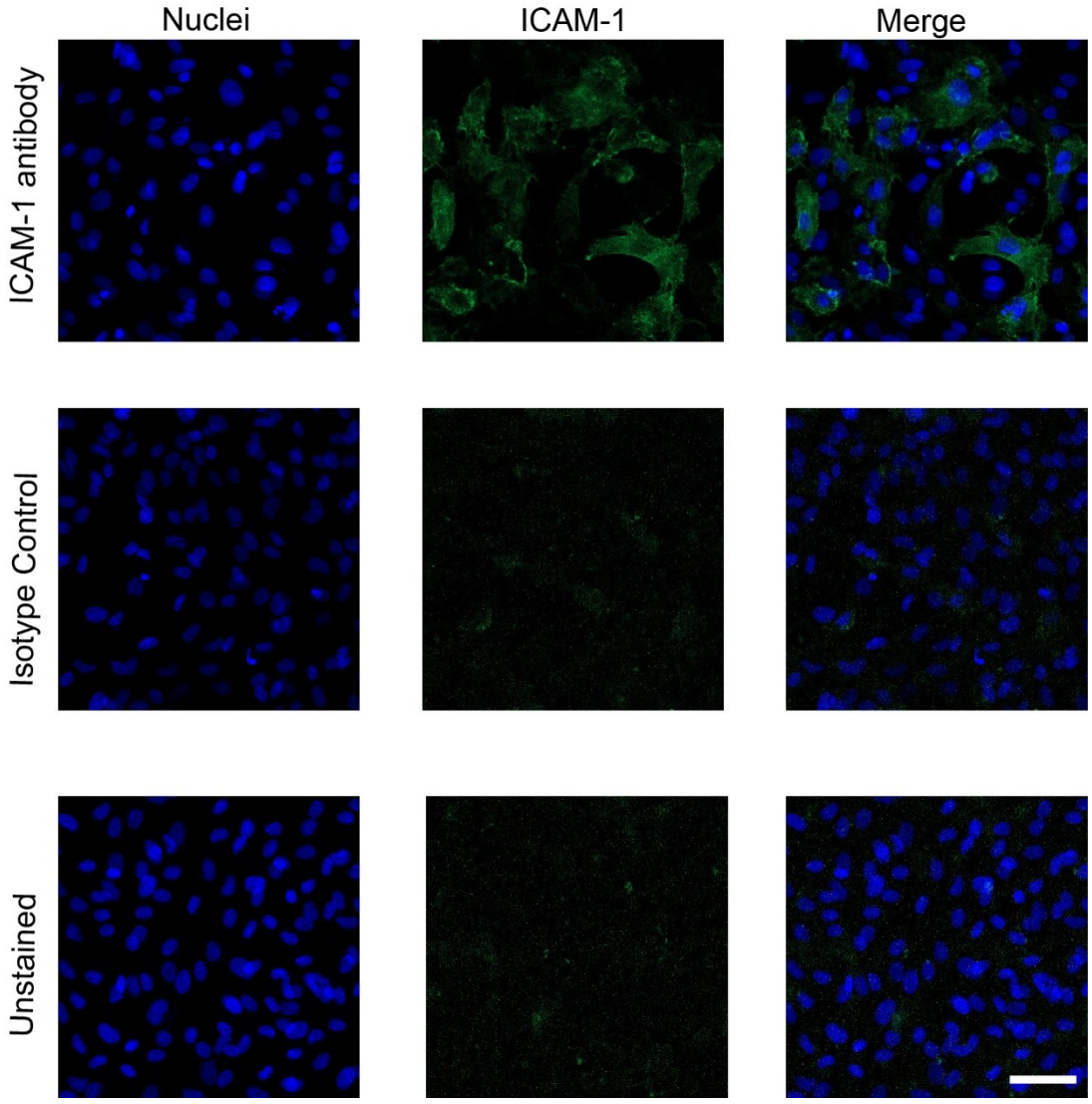

Figure S2- Isotype validation of ICAM-1 immunofluorescence in HCAECs. Representative confocal micrographs of HCAECs labelled with CD54 (ICAM-1) monoclonal antibody showing ICAM-1 fluorescence, mouse IgG1 kappa isotype control showing low background signals, and the unstained group showing minimal fluorescence. Nuclei were stained with Hoescht. The scale bar is 100 μm.