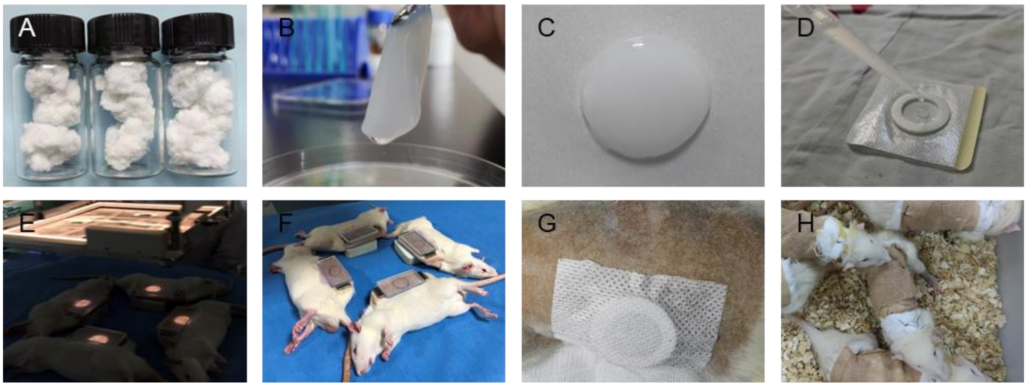

Figure 1 Preparation and application of ADM hydrogel patches in the treatment of radiation-induced skin injury

A: ADM ground into a cotton-like form, before being processed into ADM hydrogel; B: ADM Hydrogel after preparation, demonstrating sufficient mechanical strength to be lifted with forceps; C: Prepared ADM Hydrogel, retaining a circular shape; D: Uncrosslinked ADM Hydrogel being injected into the dressing, awaiting crosslinking and application to the rat skin;E-F: Rats placed on a radiotherapy machine for the induction of a radiation-induced skin injury model; four rats are positioned together for simultaneous radiation exposure; G-H: ADM hydrogel patches placed on the dorsal wounds of rats after radiation-induced skin injury.

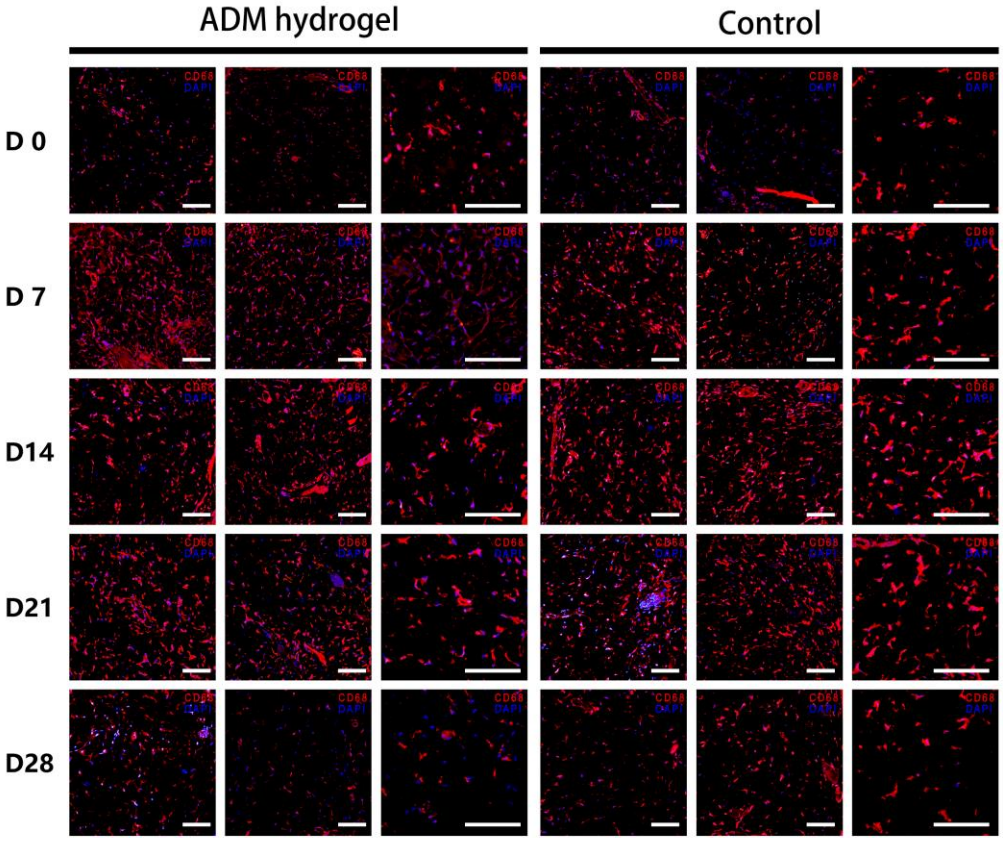

Figure 2 Representative CD68 fluorescence images. Scale bar = 50 μm

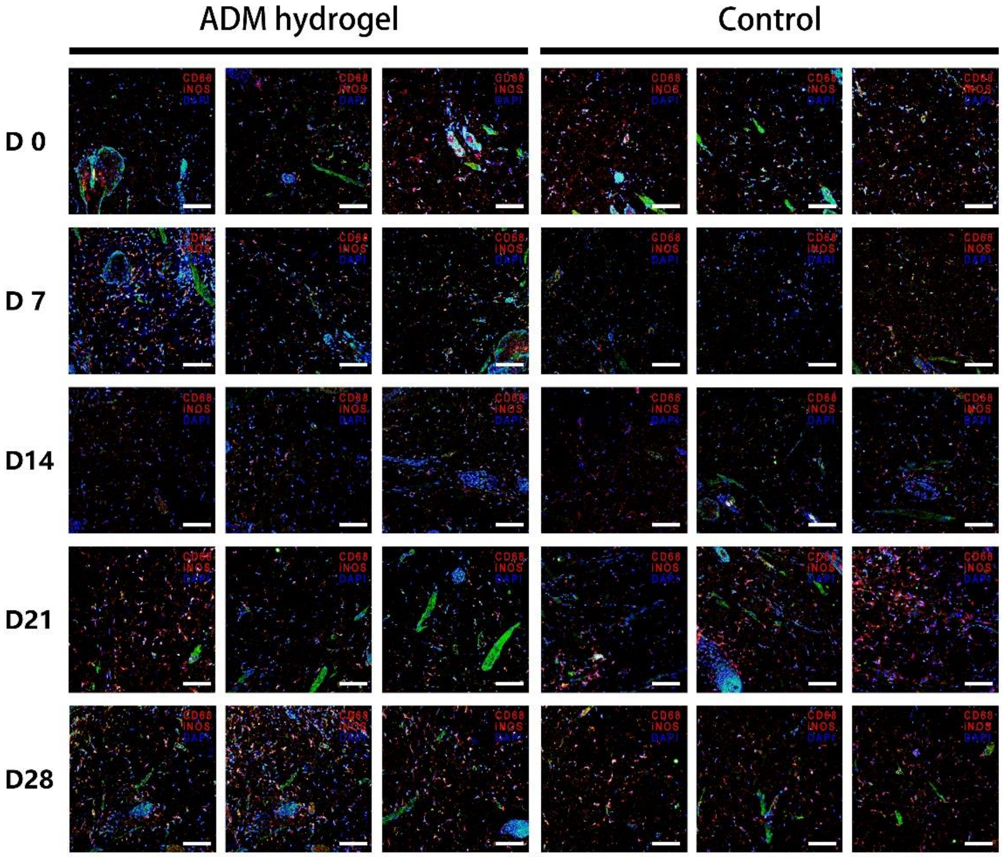

Figure 3 Representative CD68/iNOS fluorescence images. Scale bar = 50 μm

Table 1 Douglas and Fowler Radiation-Induced Skin Injury Scoring System

Score	Radiation-Induced Skin Injury Description
1.0	Normal skin
1.5	Mild erythema, slight dryness
2.0	Moderate erythema, dryness
2.5	Noticeable erythema, dry desquamation
3.0	Dry desquamation, slight keratinization
3.5	Dry peeling, keratinization, mild epidermal damage
4.0	Bullous moist desquamation, moderate erythema
4.5	Large areas of moist peeling, exudation, extensive deep keratinization
5.0	Ulceration, full-thickness skin loss

Table 2 Automated ImageJ Script for Calculating CD68 IOD

```
// Open the dialog box to select the image to be processed
filePath = File.openDialog("Choose an Image to Process");
// Open the image
open(filePath);
```

```
// Get the title of the image (i.e. the file name)
originalTitle = getTitle();
// Check if the file name ends with "-20.png" or "-40.png"
if (endsWith(originalTitle, "-20.png")) {
    // If the file name ends with "-20.png", set the scale
    run("Set Scale...", "distance=273.6667 known=20 unit=um");
    // Remove the scale area
    setTool("rectangle");
    makeRectangle(4416, 3397, 324, 143);
    run("Cut");
} else if (endsWith(originalTitle, "-40.png")) {
    // Set the scale if the file name contains "-40.png" at the end
    run("Set Scale...", "distance=546.5 known=20 unit=um");
    // Remove the scale area
    setTool("rectangle");
    makeRectangle(4143, 3395, 598, 146);
    run("Cut");
}
// Separate channels
run("Split Channels");
// Turn off the green and blue channels of the image
selectImage(originalTitle + " (green)");
close();
selectImage(originalTitle + " (blue)");
close();
// Select the red channel
selectImage(originalTitle + " (red)");
// Apply the default threshold, you can adjust the threshold manually as needed
setAutoThreshold("Default dark");
setAutoThreshold("Yen dark");
// Take a measurement
run("Measure");
// Close the image
close();
```

Table 3 Automated ImageJ Script for Batch Processing of CD68 and iNOS Colocalization Analysis

```
// Select the folder containing all images
dir = getDirectory("Choose a Directory");
// Get all file names in the folder
list = getFileList(dir);
// Define variables to store the status of red and green channel images
red_opened = false;
green_opened = false;
```

```
// Store the current processing group number
current_group = "";
// Iterate over all files
for (i = 0; i < list.length; i++) {
    fileName = list[i];
    // Find the index of the last "-", the group number is from the start of the file name to just before the last "-"
    lastDashIndex = lastIndexOf(fileName, "-");
    // Ensure lastDashIndex is valid
    if (lastDashIndex > 0) {
        // Extract group number, it is from the start of the file name to just before the last "-"
        group = substring(fileName, 0, lastDashIndex);
    } else {
        // If the file name format is incorrect, skip this file
        continue;
    }
    // If this is a new group, reset flags
    if (group != current_group) {
        red_opened = false;
        green_opened = false;
        current_group = group;
    }
    // Determine whether the current image is for red channel (CD68) or green channel (iNOS)
    if (endsWith(fileName, "-red.tif")) {
        // Open red channel (CD68)
        open(dir + fileName);
        rename("red");
        // Convert to 8-bit image
        run("8-bit");
        // Set threshold and binarize
        setAutoThreshold("Default");
        run("Convert to Mask");
        run("Invert"); // Set positive regions to white
        // Save binary image of red channel
        run("Duplicate...", "title=red_binary");
        // Mark that red channel has been opened
        red_opened = true;
    }
    if (endsWith(fileName, "-green.tif")) {
        // Open green channel (iNOS)
        open(dir + fileName);
        rename("green");
        // Convert to 8-bit image
        run("8-bit");
        // Set threshold and binarize
```

```
          setAutoThreshold("Default");
          run("Convert to Mask");
            run("Invert");
            // Save binary image of green channel
            run("Duplicate...", "title=green_binary");
        // Mark that green channel has been opened
        green_opened = true;
    }
    // When both channel images are opened, calculate colocalization area
    if (red_opened && green_opened) {
        // Use Image Calculator to find colocalization area (AND operation)
        selectWindow("red_binary");
        selectWindow("green_binary");
        imageCalculator("AND create", "red_binary", "green_binary");
        rename(group + "-colocalization");
        // Calculate area of colocalization region
        selectWindow(group + "-colocalization");    // Use new window name
        run("Measure");
        // Reset flags for processing next group
        red_opened = false;
        green_opened = false;
        // Close all open images
        close("*");
    }
}
// Notify user when done
print("Processing complete!");
```