```
% compute_pattern_properties.m

function [r_meanA, N_A] = compute_pattern_properties(A,
A_threshold_scaler)

% PRE: A is matrix of the substance concentrations,
A_threshold_scaler
% determines the factor by which the threshold (which determins
the binary
% value of a cell which used in the algorithm) is scaled.
% POST: r_meanA and N_A are the estimated mean radius and number
of patches (assuming that the
% pattern is patchy)

[m,n] = size(A);
A_thr_scaled=mean(mean(A))*A_threshold_scaler;

A_bin = A > A_thr_scaled;


sum_A_bin = sum(sum(A_bin));


% compute difference part
sum_DA_bin = 0;

A_bin_plus = [A_bin; A_bin(1,:)];
A_bin_plus = [A_bin_plus zeros(m+1,1)];
A_bin_plus(1:m,n+1)=A_bin(1:m,1);

sum_DA_bin = sum_DA_bin + sum(sum(abs(diff(A_bin_plus))));
sum_DA_bin = sum_DA_bin + sum(sum(abs(diff(A_bin_plus'))));

r_meanA = 2* sum_A_bin/sum_DA_bin;

N_A = (sum_DA_bin*sum_DA_bin)/(4*pi*sum_A_bin);

end
```