```matlab
function Q = euler_rot(alpha,beta,gamma,seq)
%EULER_ROT Euler Rotation Matrix for any rotation sequence
%   Inputs are:
%   alpha  :a scalar alpha angle in rad
%   beta   :a scalar beta angle in rad
%   gamma  :a scalar gamma angle in rad
%   seq    :a string rotation sequence (i.e. '313' for classical sequence
%           or '123' for yaw-pitch-roll)
%
%   Output is:
%   Q      :a numeric array of 3x3 direction cosine matrix

    arguments
        alpha {mustBeNumeric, mustBeReal}
        beta {mustBeNumeric, mustBeReal}
        gamma {mustBeNumeric, mustBeReal}
        seq  char
    end

    angles = [gamma,beta,alpha];% angle matrix in operation sequence
    Q_cell = cell(1,3);

    for i = 1:3
        angle = angles(i);
        switch seq(i)
            case '1'
                Q_cell{i} = rot_1(angle);
            case '2'
                Q_cell{i} = rot_2(angle);
            case '3'
                Q_cell{i} = rot_3(angle);
        end
    end

    Q = Q_cell{1}*Q_cell{2}*Q_cell{3};% Direction Cosine Matrix

    function Ra_b = rot_1(ang)
        Ra_b = [1,0,0;0,cos(ang),sin(ang);0,-sin(ang),cos(ang)];
    end

    function Ra_b = rot_2(ang)
        Ra_b = [cos(ang),0,-sin(ang);0,1,0;sin(ang),0,cos(ang)];
    end

    function Ra_b = rot_3(ang)
        Ra_b = [cos(ang),sin(ang),0;-sin(ang),cos(ang),0;0,0,1];
    end

end
```