```
% Pierce Zhang, CMOR220, Fall 2023, Projecr 8: Cryptography
% Cryptography.m
% Demonstrate the Metropolis algorithm for finding a key to break a cipher
% Last Modified: October 30, 2023

% Driver to run the methods
function Cryptography
    % read the text
    text = fileread('encodedtext.txt');
    text=strip(text);
    y = decoder(text, 10^4);
    disp('Message:');
    % decode the text
    downlowinv(y(downlow(text)))
end

% Input: character, a string
% Output: num_test, doubles representing the letter of the alphabet of each
% char of character
function [num_text] = downlow(character)
    % translate characters ('abc...z') to numbers (1,2,3,...,27)
    num_text = double(character) - 95;
end

% Input: num_text, a vector of doubles representing the letter of the
% alphabet of chars
% Output: character, the string representation of the above which is given
% by the inverse of downlow; all ` have been replaced by ' '
function [character] = downlowinv(num_text)
    % translate numbers (1,2,3,...,27) to characters ('abc...z')
    character = char(num_text + 95);
    character = strrep(character, '`',' ');
end

% Input: - text, a string to use to decipher a key
% - maxiter, number of iterations to run
% Output: y, the  best key from the Metropolis method evolution
function [y] = decoder(text, maxiter)
    load('letterprob.mat');
    M = letterprob;
    y=randperm(27);
    % apply Metropolis Algorithm to get the best y
    for i = 1:maxiter
        R = randi([1,27],1,2);
        ymaybe = repmat(y,1);
        ymaybe(R(1)) = y(R(2));
        ymaybe(R(2)) = y(R(1));

        eval_orig = loglike(downlow(text), y, M);
        eval_maybe = loglike(downlow(text), ymaybe, M);
        if (eval_maybe > eval_orig)
            y = ymaybe;
```

```
        else
            prob = exp(eval_maybe - eval_orig);
            test = rand();
            if (test <= prob)
                y = ymaybe;
            end
        end
    end
end

% Input: - text, the given text to analyze for letter pair PWCM analysis on
% M
% - guess, the guess key value to use to decipher M
% - M, the PWCM for English letter pair frequencies
% Output: value, the value of the loglike comparison given by the formula
% in the spec to evaluate the quality of the guess
function [value] = loglike(text, guess, M)
    % compute log - likelihood function
    value = 0;
    for k = 1:length(text)-1
        value = value + log10(M(guess(text(k)),guess(text(k+1))));
    end
end
```

Message:

ans =

'ever since computers there have always been ghosts in the machine random segments of code that have grouped together to form unexpected protocols unanticipated these free radicals engender juestions of free will creativity and even the nature of what we might call the soul why is it that when some robots are left in darkness they will seek out the light why is it that when robots are stored in an empty space they will group together rather than stand alone how do we explain this behavior random segments of code or is it something more when does a perceptual schematic become consciousness when does a difference engine become the search for truth when does a personality simulation become the bitter mote of a soul'

Published with MATLAB® R2023a