
Table of Contents

.....	1
.....	2
INPUT VALIDATION	2
.....	2
INITIALIZATION	2
.....	2
CALCULATIONS	2
.....	3
FORMATTED TEXT/FIGURE DISPLAYS	3
.....	3
COMMAND WINDOW OUTPUT	3
.....	3
ACADEMIC INTEGRITY STATEMENT	3

```
function v0Vals = M3_Algorithm_001_30(enNum);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ENGR 132
% Program Description
% This program will use a method similar to Algorithm 1 of M2 in order
% to calculate v0 constants for a given enzyme at 10 different
% substrate
% concentrations.
%
% Function Call
% v0Vals = M3_Algorithm_001_30(enNum);
%
% Input Arguments
% enNum - the number of the enzyme being examined as integer 1 - 5.
% Other
% values will throw an error.
%
% Output Arguments
% v0vals - an array containing all 10 v0 values for each substrate
% conc
%
% Assignment Information
% Assignment: Milestone 3, Algorithm
% Team member: Surya Manikhandan, smanikha@purdue.edu
%              Julius Mesa, jmesa@purdue.edu
%              Alex Norkus, anorkus@purdue.edu
%              Luming Lin, lin971@purdue.edu
% Team ID: 001-30
% Academic Integrity:
% [] We worked with one or more peers but our collaboration
% maintained academic integrity.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

INPUT VALIDATION

```
inval = 0; % this flag value will hold whether or not any of the
params are invalid

if((floor(enNum) ~= enNum) | (enNum > 5) | (enNum < 0)) % check if
width is a positive integer
    fprintf(2, "ERROR: enNum parameter must be an integer between 1-5
inclusive\n");
    inval = 1; % toggle flag
end

if(inval) % quit if any parameter is invalid
    return;
end

Not enough input arguments.

Error in M3_Algorithm_001_30 (line 35)
if((floor(enNum) ~= enNum) | (enNum > 5) | (enNum < 0)) % check if
width is a positive integer
```

INITIALIZATION

```
fileName = "Data_nextGen_KTesting_allresults.csv"; % the name of the
datafile

% import data vals crucial to the calculation
timeAxis = readmatrix(fileName, "range", "A4:A");
concentrationData = readmatrix(fileName, "range", "B5:CW7488");

% compute the starting column of the data given the enzyme number
origColumn = 1 + (20 * (enNum - 1));
dupeColumn = 11 + (20 * (enNum - 1));
```

CALCULATIONS

```
passWidth = 17; % width for the moving average smoothing function

v0ValsTest = []; % initialize v0s to void array to begin (for orig
test)
v0ValsDupe = []; % initialize v0s to void array to begin (for dupe
test)
```

```
for product = 0:9

    % Code in next 5 lines will isolate test and dupe data without NAN
    vals
    % NOTE: BASED OFF CODE FOUND IN https://bit.ly/2OyqAQj
    testData = concentrationData(:, origColumn + product)';
    testData = testData(~isnan(testData))';

    dupeData = concentrationData(:, dupeColumn + product)';
    dupeData = dupeData(~isnan(dupeData))';

    % Find v0 values for original test and duplicate separately. Then,
    % average those v0 values to get the final result.

    % Original test
    [timeArray, dataArray] = M3_Smooth_001_30(testData, timeAxis, 17);

    % use rise/run to find the first slope value which is our v0 val
    v0 = (dataArray(2) - dataArray(1))/(timeArray(2) - timeArray(1));
    v0ValsTest = [v0ValsTest, v0];

    % Duplicate test
    [timeArray, dataArray] = M3_Smooth_001_30(dupeData, timeAxis, 17);

    % use rise/run to find the first slope value which is our v0 val
    v0 = (dataArray(2) - dataArray(1))/(timeArray(2) - timeArray(1));
    v0ValsDupe = [v0ValsDupe, v0];

end
```

FORMATTED TEXT/FIGURE DISPLAYS

COMMAND WINDOW OUTPUT

```
% final v0 array is the average of the test and duplicate data
v0Vals = (v0ValsTest + v0ValsDupe) ./ 2;
```

ACADEMIC INTEGRITY STATEMENT

We have not used source code obtained from any other unauthorized source, either modified or unmodified. Neither have we provided access to my code to another. The function we are submitting is our own original work.

end

Published with MATLAB® R2019a