
Table of Contents

.....	1
.....	2
INITIALIZATION	2
.....	2
USE REGRESSION ALGORITHM TO GET ENZYME FUNCTION	2
.....	2
CALCULATIONS OF MICHALES-MENTEN CONSTANTS & PRICE ENZYMES	2
.....	5
FORMATTED TEXT/FIGURE DISPLAYS	5
.....	5
COMMAND WINDOW OUTPUT	5
.....	5
ACADEMIC INTEGRITY STATEMENT	5

```
function M3_exec_001_30

tic
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ENGR 132
% Program Description
% this executive function will use the algorithm to automatically
% calculate
% vmax and km parameters for each enzyme using the v0 calculated
% values from
% the algorithm. Additionally, the SSE values are calculated between
% the ideal
% and expected Michales-menten curve in order to judge goodness of
% fit.
% Finally, this function will price all the enzymes. The parameters
% and the
% recommended price are printed to the command window in a neat
% manner.
%
% Note: changed or depreciated code is commented as such. New or
% unmodified
% code will remain uncommented.
%
% Function Call
% M3_exec_001_30();
%
% Input Arguments
% N/A
%
% Output Arguments
% N/A
%
% Assignment Information
% Assignment: Milestone 3
% 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.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

INITIALIZATION

```

% figureNumber = 1; % holds the number of the figure window being
% plotted [General Change - no need to keep track of figure windows
% anymore as all plots have been eliminated]

fileName = "Data_nextGen_KEtesting_allresults.csv"; % the name of the
% datafile

% import the [S] values used to produce the plots
%substrateData = readmatrix(fileName, "range", "B3:CW3"); [Change -
% xlsread is significantly faster than readmatrix]
% Category 2 - using xlsread for small data imports saves time as
% readmatrix is only good at large dataset imports
substrateData = xlsread(fileName, "B3:CW3");

```

USE REGRESSION ALGORITHM TO GET ENZYME FUNCTION

```

[b, m, SSE, SST] = M3_Regression_001_30; % call regression function to
% get function constants
fprintf("\n\n"); % move to next line for aesthetics

```

CALCULATIONS OF MICHALES-MENTEN CONSTANTS & PRICE ENZYMES

```

for enzymeNum = 1:5 % loop through all enzymes

    % use algorithm to find the v0 values
    v0Vals = M3_Algorithm_001_30(enzymeNum);

    % find the starting column of [S] values for the given enzyme

```

```

sColumn = 11 + (20 * (enzymeNum - 1));
% compute [S] for reaction velocity plots
sData = substrateData(sColumn : sColumn + 9);

% linearize [S] and v values for the linear regression through
Hanes-Wolf Method (explanation can be found in M2 exec function)
linearSData = sData; % [s] vals do not need transformation to be
linear
linearV0Array = linearSData ./ v0Vals;

% find a regression line
linearCoeffs = polyfit(linearSData, linearV0Array, 1);

% separate the slope and the intercept from the data
linearSlope = linearCoeffs(1);
linearYIntercept = linearCoeffs(2);

% calc Vmax and Km values through the Hanes-Wolf Method
(explanation can be found in M2 exec function)
% general change - In the previous M2 submission, kM was
calculated
% through the use of the y-intercept value, which was equal to
vmax/km.
% However, using this method to solve for the kM means that any
errors
% in the vmax calculation will compound over to the kM calculation.
% Therefore, we now based our calculations off the x intercept.
vMax = 1 / linearSlope;
kM = linearYIntercept / linearSlope;

% calculate SSE values between expected and actual values for raw
data
idealV = (vMax .* sData) ./ (kM + sData); % calculate ideal v0
vals using Michaelis-Menten equation
SSE = sum((idealV - v0Vals) .^ 2);

% Price the enzyme (exponential equation)
recPrice = b * 10^(m * kM);

% display the values to the command window in a neat manner
fprintf("-----\n");
fprintf("Parameters for Enzyme NextGen-%c:\n\nv0 values (uM/
min):", 'A' + (enzymeNum-1))
disp(v0Vals);
fprintf("Vmax (uM/min): %.4f | Km (uM): %.4f | SSE: %.5f \n
\nRecommended Price: $%.2f per lb\n", vMax, kM, SSE, recPrice);

% --- ALL CODE BELOW IN THIS SECTION IS DEPRECIATED ---
% Add params to the output variables [General Change - executive
function no longer has parameters, this step is unnecessary]
% vMaxArray = [vMaxArray, vMax];
% kSubMArray = [kSubMArray, kM];
% sseArray = [sseArray, SSE];

```

```
% [General Change - eliminate all plots because it is not
necessary for this assignment and provides a large speed boost
(Category 2)]
% figure displays
% figure(figureNumber);

% First, plot the raw function output for reaction velocities
% subplot(1, 2, 1);
% plot(sData, v0Vals, 'ro');
% xlabel("Substrate Concentration [S] (uM)");
% ylabel("Reaction Velocity v (uM/min)");
% title("Unaltered Data");
% grid on
% hold on
% overlay the ideal reaction velocity vector
% plot(sData, idealV, "-b");
% legend("Raw Reaction Velocity", "Michaelis-Menten Expected
Vector", "location", "south");
% hold off

% Next, plot the linearized data according to Hanes-Wolf method
% subplot(1, 2, 2);
% plot(linearSData, sData ./ v0Vals, 'ro');
% xlabel("Linearized Substrate Concentration");
% ylabel("Linearized Reaction Velocity");
% title("Linearized Data");
% grid on
% hold on
% overlay the linear regression on to that line
% plot(linearSData, (linearSlope * linearSData) +
linearYIntercept, '-b');
% legend("Linerized Reaction Velocity", "Hanes-Wolf Best Fit
Line", "location", "north");
% hold off
% title the overall figure and increment figure number
% figureTitle = sprintf("Reaction Velocity Plots for Enzyme %d
(NextGen-%c)", enzymeNum, ('A' + (enzymeNum - 1)));
% sgtitle(figureTitle);
% figureNumber = figureNumber + 1;
```

end

FORMATTED TEXT/FIGURE DISPLAYS

COMMAND WINDOW OUTPUT

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.

`toc`

`end`

Published with MATLAB® R2019a