
Table of Contents

.....	1
.....	2
INITIALIZATION	2
.....	2
CALCULATIONS & FIGURE DISPLAYS	2
.....	6
FORMATTED TEXT/FIGURE DISPLAYS	6
.....	6
COMMAND WINDOW OUTPUT	6
.....	6
ACADEMIC INTEGRITY STATEMENT	6

```
function [vMaxArray, kSubMArray, sseArray] = M3_exec_001_30();

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ENGR 132
% Program Description
% this executive function will use the algorithm to produce product
% reaction velocity plots. Exec Will automatically calculate
% parameters
% vmax and km using the calculated values from the algorithm.
%
% Function Call
% [vMaxArray, kSubMArray, sseArray] = M3_exec_001_30();
%
% Input Arguments
% N/A
%
% Output Arguments
% vMaxArray - the array containing unique vmax values for the 5
% enzymes
% kSubMArray - the array containing unique km values for the 5 enzymes
% sseArray - the array containing the SSE values for each enzyme
% compared
% to the ideal michales-mitten curve.
%
% 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; % humds the number of the fugure window being
    plotted on

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");
```

CALCULATIONS & FIGURE DISPLAYS

```
%initialize outputs as null
vMaxArray = [];
kSubMArray = [];
sseArray = [];

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.
    linearSData = sData; % [s] vals do not need transformation to be
linear
    linearV0Array = linearSData ./ v0Vals;

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

    % separete 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)
    vMax = 1 / linearSlope;
    kM = linearYIntercept * vMax;

    % 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);

    %Add params to the output variables
    vMaxArray = [vMaxArray, vMax];
    kSubMArray = [kSubMArray, kM];
    sseArray = [sseArray, SSE];

    % 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

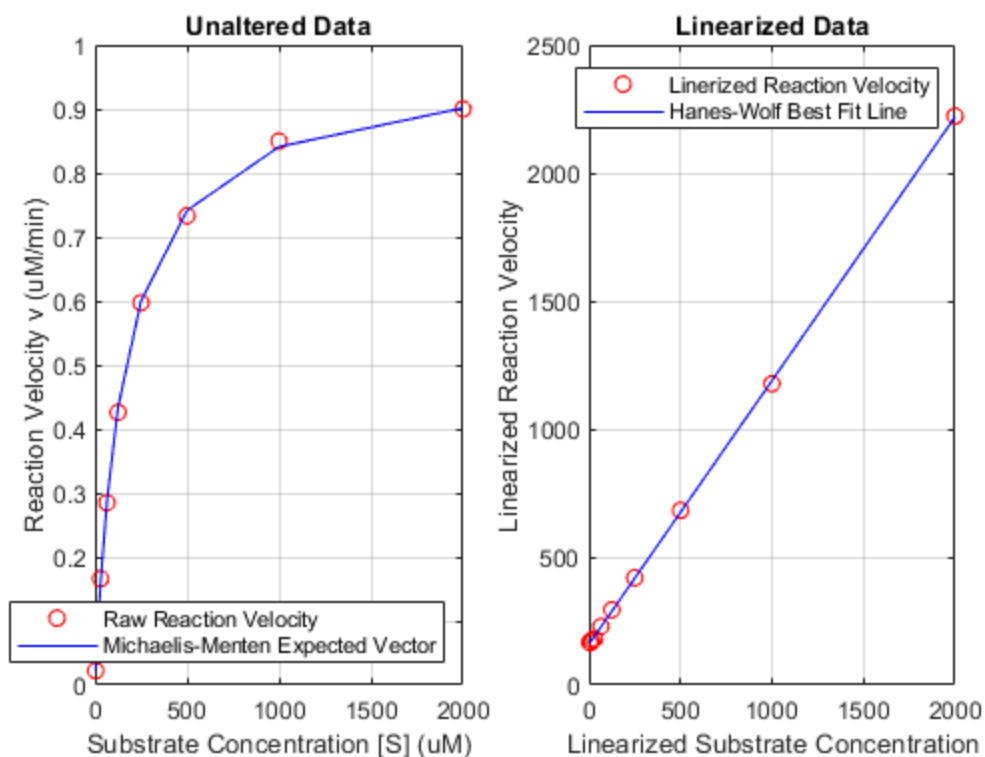
    % Next, plot the linearized data according to Hanes-Wolf method
    subplot(1, 2, 2);
    plot(linearSData, sData ./ v0Vals, 'ro');
    xlabel("Linearized Substrate Concentration");
    ylabel("Linearized Reaction Velocity");
    title("Linearized Data");
    grid on
    hold on
    % overlay the linear 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;

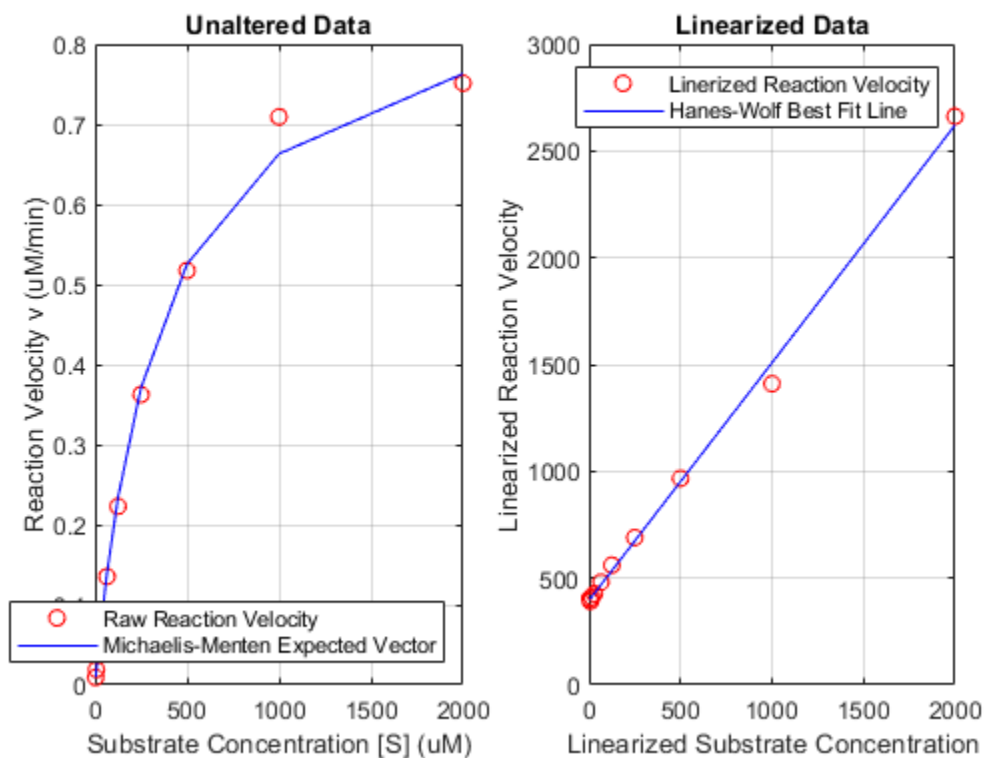
end

```

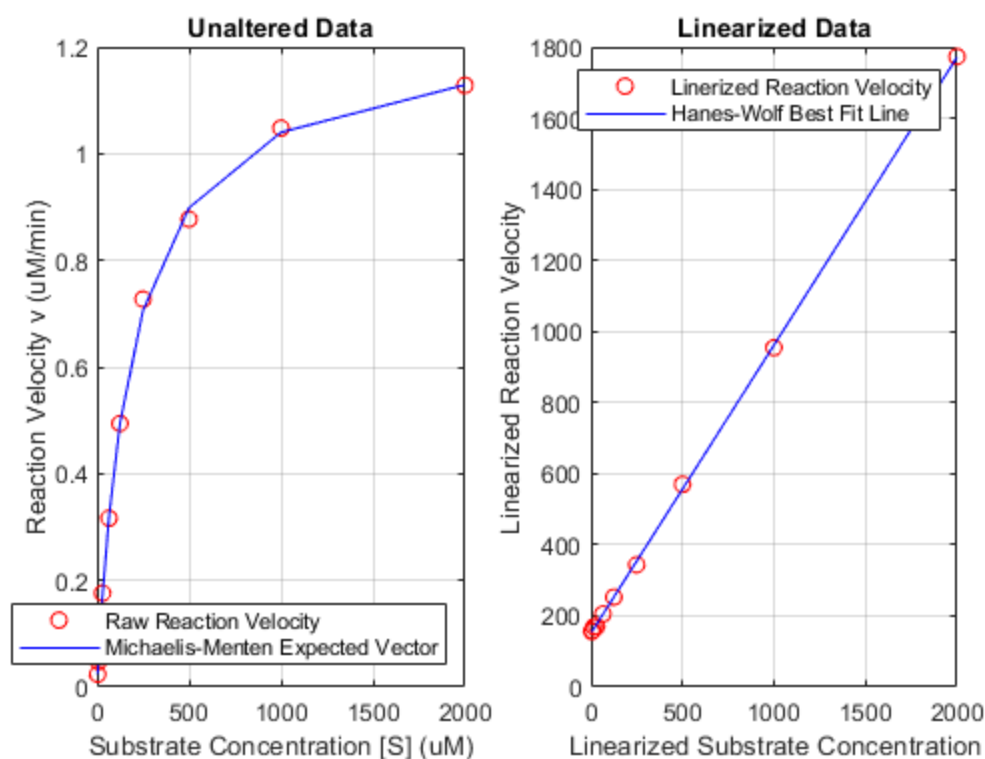
Reaction Velocity Plots for Enzyme 1 (NextGen-A)



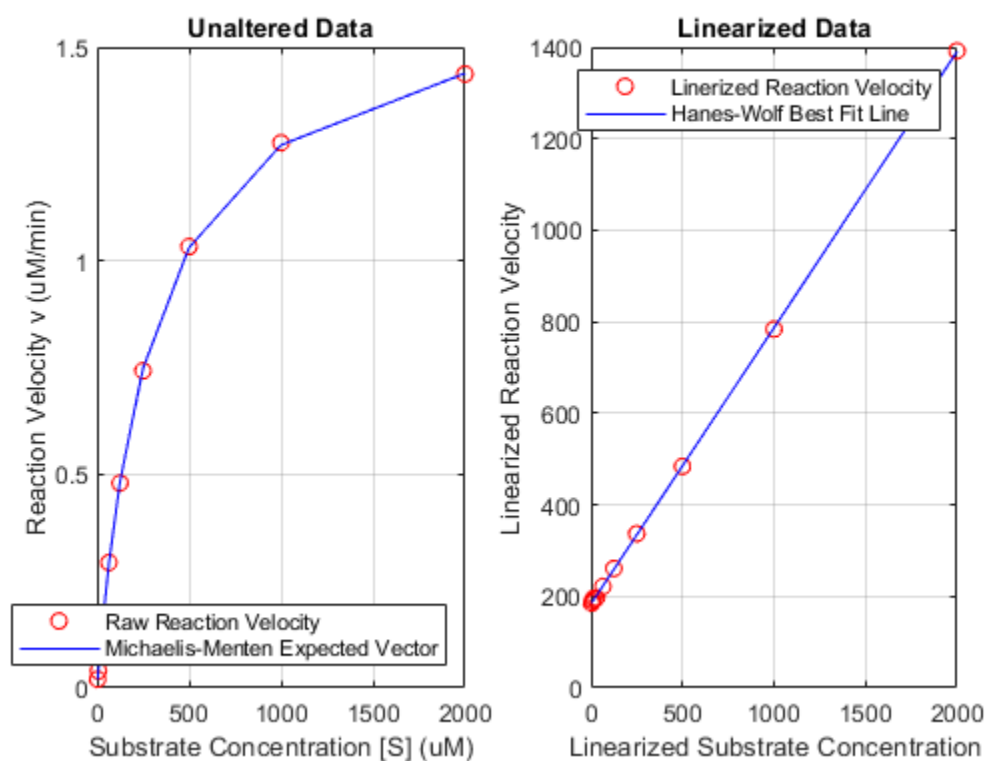
Reaction Velocity Plots for Enzyme 2 (NextGen-B)



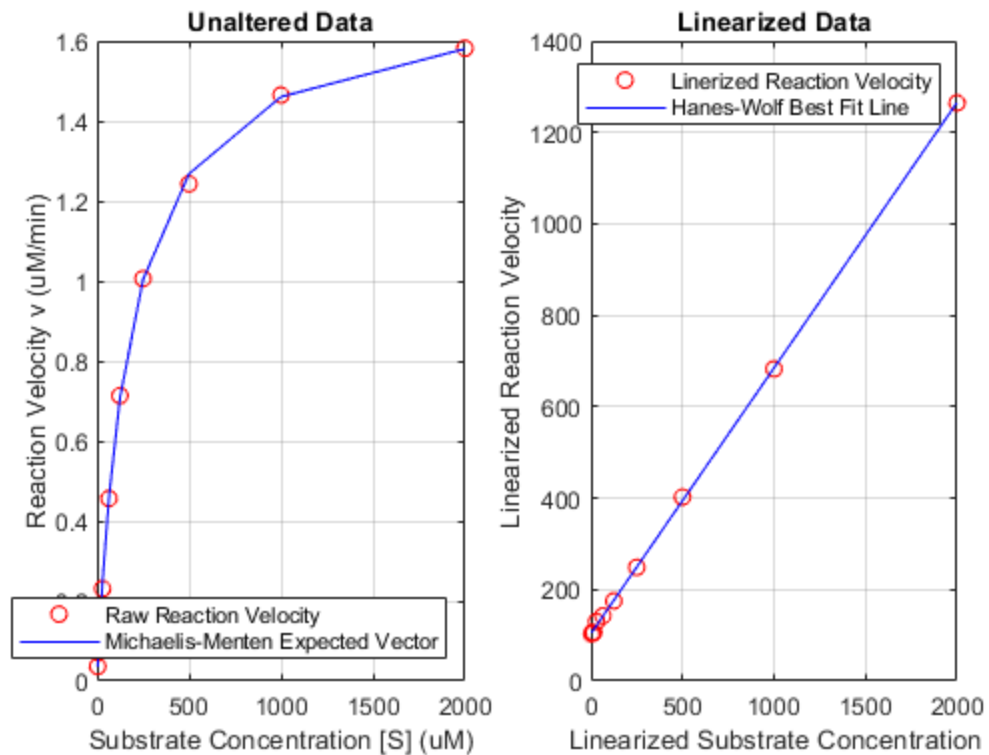
Reaction Velocity Plots for Enzyme 3 (NextGen-C)



Reaction Velocity Plots for Enzyme 4 (NextGen-D)



Reaction Velocity Plots for Enzyme 5 (NextGen-E)



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.

end

vMaxArray =

0.9714	0.8988	1.2349	1.6574	1.7222
--------	--------	--------	--------	--------

kSubMArray =

155.0399	355.4164	187.2996	304.0967	179.1636
----------	----------	----------	----------	----------

sseArray =

0.0003	0.0025	0.0010	0.0001	0.0010
--------	--------	--------	--------	--------

Published with MATLAB® R2019a