Project Milestone 4 – Technical Brief Draft

To: Avery D. Lion, President, Natural Catalysts Inc.

From: ENGR 132 Project team 002-18/ Evan Williams, Ferati Ogunwemimo, Seokjae Kim, Vivek Matta

RE: Technical Brief of Final Analysis and Results on 5 NextGen enzymes

Date: August 2, 2022

Introduction

NaturalCatalysts has asked our team to create an algorithm that will analyze enzyme data of their new 'NextGen' enzymes in order for them to be used in laundry detergents. We will analyze the enzyme data by finding the Vmax (Maximum reaction velocity) and Km (Michaelis constant) of the reaction of the enzyme with a substrate. We are constrained to writing an algorithm that can be implemented into any data processing application.

Our algorithm consists of a group of UDFs (user defined functions), the first of this finds the Vo (V naught, initial velocity of reaction) for PGO-X50 and each NextGen enzyme's test then averages any duplicates of NextGen enzyme, the second UDF uses Lineweaver-Burk linearization of a Michaelis-Menten model to find Vmax and Km, and the third UDF is the main function that combines these subfunctions in order to find Vmax, Km, and Vos for each enzyme.

Our main decision to improve the accuracy of our algorithm was in choosing the data points we used for calculating the initial velocity. Originally, we started by using 1.25% of the data for each test, but we switched to 53 data points for each test. We found this to be much more accurate as it decreased our percentage error to under 2% for every V naught, 0.54% for Vmax, and 0.28% for Km which are very low error margins.

Another decision that me made to improve the accuracy of the algorithm was using Lineweaver-Burk linearization to linearize the Michaelis-Menten plot and find Vmax and Km. We chose to use this method of linearization as it is the most common and is more accurate than other methods such as the Hanes-Woolf method. Also, according to the Science Alert, Lineweaver-Burke plot gives a more accurate estimate of Vmax (Science Alert). This decision made our algorithm more accurate as we only needed to find the reciprocal of the data as opposed to more calculations in other methods which would be less accurate due to more rounding.

Finally, a third decision that we made was improving the speed of the program. Since there was a lot of repetition in this project, we thought of ways to create a repeating structure for our code. We thought of for loops and while loops. We used the tic and toc feature to calculate the speed of the code snippet for each type of loop. In conclusion, we found that the for loop was a little bit faster than the while loop. This was necessary since we were outputting many graphs and a lot of text to the command window. Having a faster program was important to complete this task.

Parameter Identification Procedure

First, we imported data from csv file containing results of the experiments. "Data_nextGen_KEtesting_allresults.csv". Next, Using the first 45 values of product concentration([P]), for one substrate concentration([S]), use linear regression to find the slope of

the line formed by plotting those concentrations against the corresponding time. This is the initial velocity for that substrate concentration of a particular enzyme.

Repeating this process for all substrate concentrations of all enzymes gives the initial velocities of all the substrate concentrations of the original data and duplicate. Third, we calculated the 10 average initial velocities of each 5 NextGen enzyme by averaging the initial velocities of all the substrate concentrations of the original data and duplicate.

Then, linearize both the substrate concentrations and the average initial velocities using the Lineweaver-Burk method: find the reciprocal of each value (1/value). Do this for all substrate concentrations and their corresponding average initial velocities for one enzyme.

We used linear regression to find the slope and y-intercept of the line formed by plotting the linearized initial velocities against the linearized substrate concentrations. Vmax was found by finding the reciprocal of the y-intercept. Km is found by multiplying the slope by Vmax. We repeated the calculations for each process.

For the PGO-X50, we import data from "Data_PGOX50_enzyme.csv" file and found initial velocity by using the first 53 values of [P] when finding the slope. Once we had this initial velocity, finding Vmax and Km followed the same process that the 5 NextGen enzymes did.

Results

From Table 2, Enzyme D has the highest value for Vmax (1.4238 µM/s) and Km(253.25 µM) while Enzyme B has the lowest Vmax (0.99 µM/s) and Enzyme A has the lowest Km (171.27 µM)

From Table 1, the final algorithm is very efficient as the percent error of the algorithm's Km values for PG0X50 compared to known/given values is 0.28%. While for the Vmax values, percent error is 0.54% and the largest percentage error in the initial velocity calculations is 1.6%.

Interpretation

The error that we had in this process was largely due to the noise in the data. When analyzing our algorithm with known values of the enzyme PGO-X50 to test our accuracy. We found our algorithm to be very accurate with an error of 0.54% for Vmax, 0.28% for Km, and an average of 0.46% of the 10 Vos. This low of error percent shows that our algorithm was very accurate, and any error was most likely due to the noise of the data provided.

NaturalCatalysts can say that their enzyme E will perform the best. They can say this as E was the second-best enzyme in both Vmax and Km with a Vmax of 1.36 and a Km of 177.59 which will result in a fast and effective reaction. NaturalCatalysts can also say that all their enzymes are fairly similar in performance as Vmax range of 0.43 and a Km range of 82.45 which shows a high amount of consistency across the company.

References

Pant, M., Sharma, P., Radha, T., Sangwan, R.S., & Ray, U. (2008). *Nonlinear Optimization of Enzyme Kinetic Parameters.* Science Alert. Retrieved August 1, 2022, from https://scialert.net/fulltext/?doi=jbs.2008.1322.1327

Appendix: Figures and Tables

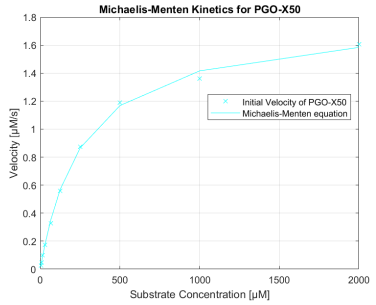

Figure 1

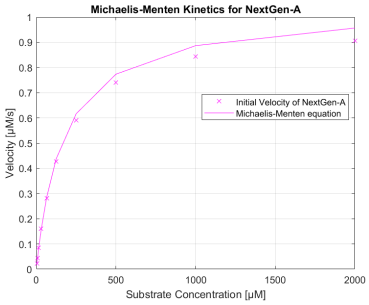

Figure 2

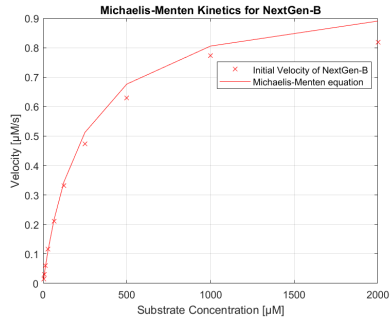

Figure 3

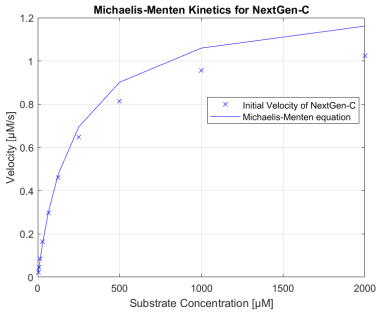

Figure 4

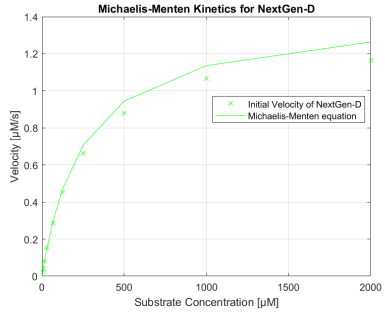

Figure 5

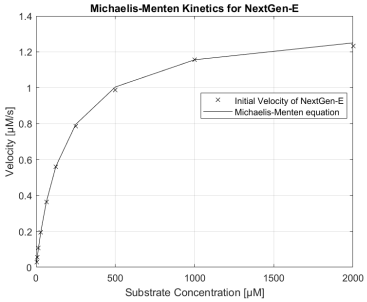

Figure 6

Parameter (µM/s)	PGO-X50 Reference Values	M3_Algorithm	Percent Error	M4_Algorithm	Percent Error
v_{0_1}	0.025	0.0248	0.8	0.0246	1.60
v_{0_2}	0.049	0.0495	1.02	0.0487	0.61
v_{0_3}	0.099	0.0993	0.30	0.0993	0.30
v_{0_4}	0.176	0.1758	0.11	0.1746	0.80
v_{0_5}	0.329	0.3281	0.27	0.3291	0.03
v_{0_6}	0.563	0.5595	0.62	0.559	0.71
v_{0_7}	0.874	0.8866	1.44	0.8737	0.03
v_{0_8}	1.192	1.2060	1.17	1.1876	0.37
v_{0_9}	1.361	1.3532	0.57	1.3598	0.09
$v_{0_{10}}$	1.603	1.6093	0.39	1.604	0.06
V_{max}	1.806	1.7959	0.56	1.7963	0.54
K_m (µM)	269.74	266.6057	1.16	268.9821	0.28

Table 1

	M3 - Enzyme Parameters		M4 - Enzyme Parameters	
	V_{max} (µM/s)	K_m (µM)	V_{max} (µM/s)	K_m (µM)
NextGen-A	1.0237	173.4885	1.0384	171.2650
NextGen-B	0.9453	227.3872	0.9954	235.7527
NextGen-C	1.2404	210.1667	1.2847	212.6062
NextGen-D	1.3673	247.7611	1.4238	253.2536
NextGen-E	1.3430	181.6917	1.3612	177.5892

Table 2