ABSTRACT

KARAM, AMANDA LOUISE. Development of Photochemical Microsensors for Evaluating Light Distributions within Microalgal Photosynthetic Bioreactors. (Under the direction of Dr. Joel Ducoste).

The use of microalgae-based biofuel feedstocks has the potential to reduce fossil fuel use and greenhouse gas emissions. Microalgae are known for their high growth rates and their ability to accumulate high levels of lipids relative to other terrestrial crops; given these benefits, considerable research has been focused on microalgae-based biofuel production processes. However, no industrial-scale microalgal bio-refineries currently exist due to the energy-intensive production processes including cultivation and microalgae/lipid separation.

Optimal design and operation of photosynthetic bioreactors (PBR) for microalgae cultivation is essential for improving the environmental and economic performance of microalgae-based biofuel production. Models that estimate the turbulent flow of fluid along with the transport of nutrients, diffusion of CO₂, transport and growth of the microalgae, and light irradiance are critical for this optimization process. Since microalgae are photosynthesizing organisms, these models should provide an accurate prediction of the overall light distribution throughout the PBR. Light distribution is difficult to measure, and thus the model predictions are difficult to validate.

This research focuses on developing a Lagrangian method for evaluating the light distribution within PBRs using novel photochemical microsensors. These microsensors were developed using microalgae-sized microspheres coated with a dye that increases in fluorescence when exposed to photosynthetically-relevant blue wavelengths of light. The dose-response kinetics of these dye-coated microspheres were established by varying known dosages of collimated light and quantifying the fluorescence response of the microsensor in a
flow cytometer on an individual particle basis. A deconvolution scheme was then used to determine the light dosage distribution using the fluorescence distribution of the microsensors. This method accurately predicted the light dosage distribution of ‘convolved’ samples made from microsensors exposed to various light dosages. The microsensors were also used to quantify the light within a flat-plate, bench-scale PBR. This novel method predicted the average light dosage within ~10% of static sensor measurements and gave insight into the distribution of the light within the PBR that was not possible to measure with static sensors. While further tests need to be performed to confirm the efficacy of these microsensors using different reactor designs and configurations, these innovative light sensors show promise for validating optimization models for designing a full-scale, microalgal-PBRs.
Development of Photochemical Microsensors for Evaluating Light Distributions within Microalgal Photosynthetic Bioreactors

by
Amanda Louise Karam

A thesis submitted to the Graduate Faculty of North Carolina State University in partial fulfillment of the requirements for the degree of Master of Science

Environmental Engineering

Raleigh, North Carolina
2016

APPROVED BY:

________________________________________  _________________________________
Dr. Joel Ducoste                    Dr. Francis de los Reyes III
Committee Co-Chair                Committee Co-Chair

________________________________________
Dr. James Levis
DEDICATION

“The world is your oyster, don’t shuck it up!”

This thesis is dedicated to the memory of my friend, Vanessa Lynn Allen… who was there for me as I began my journey into graduate school and whom I leaned on – and laughed with – through many difficult times. I wish I could tell you how excited I am to finally finish this thesis; I know you would be proud. You have been, and will continue to be, dearly missed…
BIOGRAPHY

Amanda Karam completed her Bachelor of Science with high honors in environmental science and a minor in computer science from the University of North Carolina at Chapel Hill. With pursuit to continue her passion for science, engineering, and research, she started graduate school at North Carolina State University under the advising of Dr. Ducoste and Dr. de los Reyes in the Department of Civil, Construction, and Environmental Engineering. She is thankful for the colleagues she has gotten to learn and grow from, and the invaluable training opportunities she has had at NC State. After the successful completion of her master’s degree, Amanda looks forward to advancing her knowledge and continuing her research in environmental engineering as a doctoral candidate.
ACKNOWLEDGMENTS

Funding:
National Science Foundation (NSF), Emerging Frontiers in Research and Innovation (EFRI)

Advisors, Committee, Mentors, Graduate Student Support
Dr. Joel Ducoste, Professor, NCSU Department of Civil, Construction, and Environmental Engineering (CCEE);
Dr. Francis de los Reyes, Professor, NCSU Department of CCEE
Dr. James Levis, Assistant Professor, NCSU Department of CCEE
Dr. Ranji Ranjithan, Professor, NCSU Department of CCEE
Renee Howard, Graduate Services Coordinator, NCSU Department of CCEE
Barbara Simerson, Administrator Support Specialist, NCSU Department of CCEE

Colleagues & Collaborators:
Nicholas Evans, Jing Wu, and Jiawen Liu, Undergraduate research assistants
Rohan Patil, Graduate student, NCSU Department of Chemical and Biomolecular Engineering
Jacob Dums and Colin Murphree: Graduate students, NCSU Department of Plant and Molecular Biology
Roshan Wathore, Graduate student, NCSU Department of CCEE
Catherine McMillian and Yi Chun Lai, Graduate students, NCSU Department of CCEE
Christopher Stanford, Machine Shop Supervisor, NCSU College of Engineering
Jake Rhodes, Machine Shop Supervisor, NCSU Department of CCEE
Sarah Schuett, Flow Cytometry and Cell Sorting Laboratory Director, NCSU College of Veterinary Medicine
Dr. Jan Genzer, Professor, NCSU Department of Chemical and Biomolecular Engineering
Dr. Amy Grunden, Professor, NCSU Department of Plant and Microbial Biology
Dr. Heike Sederoff, Professor, NCSU Department of Plant and Microbial Biology
Dr. Kyle Gee, Senior Director, R&D, Cellular Analysis, ThermoFisher Scientific (formerly Life Technologies Corporation)
Dr. Vladimir Belov, Co-founder, Abberior GmbH, Inventor of Abberior GmbH caged and photo-switchable dye series
# TABLE OF CONTENTS

List of Tables .......................................................................................................................... vii
List of Figures .......................................................................................................................... viii
Chapter 1 - Introduction ......................................................................................................... 1
  Objectives .............................................................................................................................. 4
Chapter 2 - Literature Review & Relevant Background ...................................................... 5
  Past work with microsensors as non-biological surrogates ............................................... 5
  Microsensor development and relevant background ........................................................... 6
    Microsphere selection .......................................................................................................... 7
    Photo-reactive dye selection .............................................................................................. 9
    Measuring principle: flow cytometry overview ................................................................. 11
Chapter 3 - Methods ............................................................................................................. 14
  Microsensor development ................................................................................................. 14
  Quantifying dosage-response kinetics of microsensor ....................................................... 17
    Collimated beam tests and setup ....................................................................................... 17
    Sample analysis in flow cytometer .................................................................................. 18
    Signal processing microsensor fluorescence data ............................................................ 21
    Dosage-response kinetics experimental procedure .......................................................... 26
  Proof-of-concept testing ..................................................................................................... 27
    Predicting a known light distribution with deconvolution .............................................. 27
    Predicting the light distribution within a bench-scale PBR ............................................. 28
Chapter 4 - Results & Discussion .......................................................................................... 33
  Quantifying microsensor dosage-response behavior and kinetics ..................................... 33
    Distribution fitting ........................................................................................................... 33
    Dose-response parametric equations .............................................................................. 34
    Reciprocity ....................................................................................................................... 36
  Proof-of-concept testing ..................................................................................................... 36
    Predicting a known light dosage distribution .................................................................. 36
    Predicting the light distribution of bench-scale PBR ...................................................... 40
Chapter 5 - Considerations, Future Work, & Conclusions ................................................. 51
  Deconvolution and light dosage distribution estimates ....................................................... 51
Proof-of-concept testing and validation ................................................................. 53
Estimating light intensity within a PBR ............................................................... 53
PBR experimental testing conditions .................................................................. 53
Summary and future applications ........................................................................ 54
References ............................................................................................................ 56
Appendix .............................................................................................................. 61

Section 1: Supplemental tables and figures .......................................................... 62
Table A1-1: Supplemental coupling information for labeling dye with microspheres.
Experimental reference for coupling dye to beads. Many calculations/values in table
below are based on specifications as laid out in Table 3-1, Table 3-2, and Table 3-3... 62
Figure A1-1: Spectral power distribution from blue LED lamp (LumiGrow) used in all
experiments. Spectral light intensity of light used for testing microsensors. The total
light intensity was calculated by integrating the spectral power distribution between 400
to 500 nm .............................................................................................................. 62
Figure A1-2: Absorbance and emission (caged and uncaged) spectrum for Abberior
CAGE-552 NHS-ester dye ..................................................................................... 64

Section 2: MATLAB supplemental information .................................................... 65
Section 2-2: Kernel density estimates for fluorescent distributions ......................... 65
Section 2-1: Distribution testing with AllFitDist .................................................... 66

Section 3: Python 3.2.1 Scripts for deconvolution .............................................. 67
Script 1: Deconvolution without interpolation ..................................................... 67
Script 2: Deconvolution with interpolation: ........................................................ 71
Function 1: get_convolution_and_dose_matrix_and_output_file ......................... 75
Function 2: get_output_file_for_no_interpolation .............................................. 77
Function 3: get_output_file_for_interpolation ................................................... 79
Function 4: main_function_no_interpolation .................................................... 81
Function 5: main_function_interpolation .......................................................... 85
Function 6: objective_function ........................................................................... 89
LIST OF TABLES

Table 3-1 Specifications for 3µm amine-surface labeled microspheres.......................... 16
Table 3-2 Specifications for Abberior Cage-552 (NHS-Ester) caged dye. ......................... 16
Table 3-3 Specifications for Quantabio design m-dPEG®₂₄-NHS polymer....................... 16
Table 3-4 Overview of experimental conditions tested in collimated beam setup............. 26
Table 4-1 BIC criteria values used to evaluate best fit for stable parameter curves.......... 35
Table 4-2 Light intensity measurements taken across PBR for proof-of-concept testing. Light measurements across the reactor depth, without water, taken with handheld spectrometer. Note: Letter references refer to Figure 3.11................................................................. 43
Table 4-3 Deconvolution prediction for the average light dosage from the PBR and collimated samples as compared to measured average light dosage estimates. Deconvolution predictions made with dosage bin widths of 1, 0.5, 0.25 kJ-m-2. *Indicates that the deconvolution predictions for the collimated samples fall within a 95% confidence interval of the spectrometer-based measurements. Note: Confidence intervals for the measured average light dosage in PBR were not computed due to measurement limitations. ............................ 44
LIST OF FIGURES

Figure 1.1  Three common PBR reactor design with theoretical light distributions shown: tubular (LEFT), flat-plate (MIDDLE), and bubble-column (RIGHT). ...................................................... 2

Figure 2.1  Microsensor overview diagram. A simplified overview of microsensor showing how 1) prior to light exposure, the photo-sensitive dye attached to microsphere will have little to no fluorescence when measured in a flow cytometer, 2) during light exposure, the dye reacts with light, uncaging fluorophore, and 3) after light exposure, the microsensor will increase in light intensity. .................................................................................................................. 6

Figure 2.2  Images of 10 µm microspheres and *Dunaliella viridis* microalga cell............... 8

Figure 2.3  Microscopic image of 10 µm amine-surface modified microspheres, noting minor aggregation (ARROWS) .................................................................................................................. 9

Figure 2.4  Absorbance spectrum of photosynthesizing pigments in marine microalgae *Dunaliella viridis*, as measured in spectrometer. Peak absorbance occurs around 436 nm... 10

Figure 2.5  Flow cytometry conceptual diagram. Illustration of internal components of flow cytometer showing hydrodynamic focusing and laser, optical detectors used for measuring individual particle characteristics. .............................................................................................................................. 12

Figure 3.1  Visual evidence of aggregation of dyed microspheres. A) dyed-coated microspheres without polymer, B) dyed-coated microspheres with polymer, C) 0.22 µm filtered DI water as control for comparison .......................................................... 15

Figure 3.2  Diagram of collimated beam apparatus used to expose microsensor samples to uniform light source .................................................................................................................................. 17

Figure 3.3  Custom 3-D printed spectrometer holder used to position handheld spectrometer for measuring the light and light uniformity across the petri dish at 0.5 cm increments........ 18

Figure 3.4  Flow cytometry density plots showing gating technique used to select consistent, non-aggregated microsphere populations. These gates were chosen based on A) FSC-H vs FSC-A, or relative particle size height vs area, Gate #1, and B) SSC-H vs SSC-A or relative particle scatter/granularity height vs area, Gate#2 .............................................................................................................................. 20

Figure 3.5  Fluorescence of microsensors over range of dosages. These kernel density estimates of the fluorescence show overlapping fluorescence values and increasing variance with increasing dosage ........................................................................................................ 21

Figure 3.6  Example of interpolated curves describing the distribution of microsensor fluorescence for different exposures to a collimated light source ......................................................... 25

Figure 3.7  Convolved sample preparation diagram showing how a bimodal distribution sample would be prepared using 5 distinct dosages if the microsensor number concentration were assumed constant for all five dose components. ........................................ 28

Figure 3.8  Bench-scale PBR used for proof-of-concept testing. A) reactor as setup for proof-of-concept tests, and B) schematic drawing given in inches ........................................... 28

Figure 3.9  Dye test used to determine characteristic mixing time of bench-scale reactor. Pictures show dye dispersion at A) 0, B) 15, C) 30, and D) 120 seconds for this bench-scale PBR with axial impeller mixing at ~60-80 RPMs ........................................................................ 29

Figure 3.10  Positions across PBR at which light intensity measurements were taken with a handheld, visible light spectrometer. A, B, C, F, and G were used to center the PBR body. B, D, E were used to estimate the average light intensity in the PBR; these values were assumed to be representative of the average light within components #1, #2, #3 respectively. ............ 31
Figure 4.1 BIC criteria values for most promising distributions for describing the distribution of the microsensor fluorescence after exposure to known, uniform light dosage. Values represent average BIC across all samples replicates and dosages. Note: Distribution fitting analysis was done for each sample, across all dosages and replicates; the stable distribution consistently had lowest BIC. .................................................................................................................. 33

Figure 4.2 Empirical versus stable fitted fluorescence distribution for microsensors over range of light dosages experimentally measured in collimated beam apparatus. ................................. 34

Figure 4.3 Stable distribution parameters versus light dosage. A) $\alpha$ -stability, B) $\beta$ - skewness, C) $c$ -scale, and D) $\mu$-location parameter versus light dosage under a 1330 $\mu$watt-cm$^{-2}$ (PINK) and 1880 $\mu$watt-cm$^{-2}$ (GREEN) collimated light intensity. ........................................... 35

Figure 4.4 Fluorescence signal output as measured by flow cytometer and shown as a kernel density function (BLUE) as compared to signal output estimated with deconvolution algorithm (ORANGE) for convolved samples with A) even, B) unimodal, C) bimodal, and D) skewed dosage distributions. ................................................................................................. 37

Figure 4.5 Measured vs deconvolution predictions for light dosage distribution for convolved samples. Estimates predicted based on concentration and volume (LEFT) and deconvolution (RIGHT) for A) even, B) unimodal, C) bimodal, and D) skewed dosage distributions. Error bars represent standard error for $n = 3$ samples. ................................................................. 39

Figure 4.6 Fluorescence distribution curves from experimental samples after 1, 2, 4, and 8 minutes of light exposure within proof-of-concept PBR. Fluorescence distributions are shown as smoothed kernel density functions based on flow cytometry fluorescence output. 40

Figure 4.7 Predicted fluorescence distribution curves generated by interpolation of stable distribution parameters compared to fluorescence distribution curves generated experimentally via collimated beam testing. Interpolated curves are shown for light dosages between 1 and 20 kJ-m$^{-2}$ at a light dosage bin width of 1 kJ-m$^{-2}$ .......................................................... 41

Figure 4.8 Average light dose distribution predictions for PBR at 1, 2, 4, and 8 minutes of light exposure using an interpolated light dosage bin width of A) 1, B) 0.5, and C) 0.25 kJ-m$^{-2}$. Error bars represent one standard deviation in average dosage prediction after $n = 100$ optimization trials. ........................................................................................................... 42

Figure 4.9. Light dosage predictions for samples exposed to a known, collimated light source using deconvolution with an interpolated dosage bin width of A) 1, B) 0.5, and C) 0.25 kJ-m$^{-2}$. Error bars represent the standard deviation of solutions after $n = 100$ optimization trials. 46

Figure 4.10 Graphical visualization of the impacts of fitting/interpolation error on deconvolution results when analyzing the 1.12 kJ-m$^{-2}$ collimated sample. Deconvolution attempts to reconstruct, as shown in RED, the fluorescence distribution of the 1.12 kJ-m$^{-2}$ experimental sample (PURPLE) using the interpolated curves (BLUE). For the 1.12 kJ-m$^{-2}$ sample, the deconvolution algorithm can achieve a smaller minimum – the RSS between the reconstructed (RED) and the actual (PURPLE) fluorescence distributions – by assuming a small fraction of the experimental sample was exposed to light dosages in the 4-6 kJ-m$^{-2}$ range. The predicted dosage distribution is shown as percentages next to interpolated curves. .................................................................................................................. 47

Figure 4.11 Residual sum of square (ORANGE) and the percent difference between spectrometer-based estimates for average light dosage and deconvolution predictions (BLUE) at three light dosage bin widths (0.25, 0.5, 1.0 kJ-m$^{-2}$). This graphic highlights one implication of overfitting in deconvolution, i.e. decreasing the bin width can lead to an overall lower RSS for optimization but a higher error from true value. .................................................................................................................. 48
Figure 4.12 BIC (LINE) and RSS (COLUMN) values for dosage distribution predictions for samples after 1, 2, 4, and 8 minutes of light exposure within PBR. Predictions were made using deconvolution where bin width for interpolated fluorescence distribution curves were varied from 1 (ORANGE), 0.5 (BLUE), and 0.25 (GREEN) kJ-m\(^{-2}\).
CHAPTER 1 - INTRODUCTION

Since the beginning of the 20th century, petroleum-based fuel has been used to power transportation systems. However, 21st century concerns regarding global warming and finite fossil fuel resources has encouraged research and development of renewable transportation fuels. In response, governments worldwide have developed policies that reduce the consumption of petroleum-based fuels and encourage the production of renewable fuel sources\textsuperscript{1,2}. For example, the United States Environmental Protection Agency (EPA) has created the Renewable Fuel Standard, a program that requires 36 billion gallons of the annual 140 billion gallons of fuel mix to come from renewable fuel sources by 2022\textsuperscript{2}.

Chisti (2007) reported that 36 billion gallons of fuel from existing biofuel feedstocks is not realistic with first-generation biofuels. Biofuels from microalgae is a promising alternative fuel source\textsuperscript{3,4} as microalgae-based fuels hold several advantages to first-generation biofuels based on land plants\textsuperscript{5} (e.g. corn and soybeans) and second-generation biofuels based on cellulose from non-edible plant biomass\textsuperscript{4}. Unlike first-generation biofuels, microalgae-based biofuels divert fewer resources away from the food, land, and water supply since microalgae can be cultivated year-round on barren land using salt water or wastewater as the liquid media\textsuperscript{4,6}. Thus, this renewable fuel source may also help remove excess nutrient pollutants in water, offsetting energy and greenhouse gas (GHG) emissions that would be required to treat wastewater. Microalgae are also known for their high growth rates – surpassing the productivity of even the fastest growing terrestrial crops\textsuperscript{5} – and their ability to accumulate high levels of neutral lipids. These high energy storage compounds, particularly triacylglycerol, can be converted into biodiesel\textsuperscript{4,7}. Currently, no industrial-scale microalgae-to-biofuel plant exists due to the high costs from the energy-intensive production processes,
which consists of microalgal cultivation, harvesting/lipid separation, and lipid refining into biodiesel\textsuperscript{4,8–11}.  

Microalgae cultivation sets the groundwork for all downstream processes, making research and advances in this area paramount. Already, many researchers have suggested that photosynthetic bioreactors (PBRs), or clear, enclosed installations for the production of phototrophic microorganisms in an artificial environment, are superior to open pond systems for growing microalgae for biofuels since they offer increased control and versatility\textsuperscript{4,12,13}. Figure 1.1 displays three basic PBR designs, tubular, flat-plate, and bubble-column, each with unique geometries and energetic costs.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{Three common PBR reactor design with theoretical light distributions shown: tubular (LEFT), flat-plate (MIDDLE), and bubble-column (RIGHT).}
\end{figure}

Past research has suggested that to achieve high cell densities the PBR thickness should be as small as possible to minimize dark zones\textsuperscript{14}, i.e., a high surface-to-volume ratio (SVR).
High SVR, however, leads to higher material costs and frictional losses from pumping water through narrow pathways. Therefore, low-SVR designs have been considered preferable from a life cycle assessment standpoint but due to large dark portions, still lack the productivity required to make this production process economical\textsuperscript{13,15}. These assessments are only as reliable as their validation data. Difficulties often arise in quantifying the productivity of reactor designs due to lack of industrial data, variation and quantification of microalgal growth kinetics and reactor design parameters. Powerful mathematical models that consider turbulent fluid flow along with the transport of nutrients, diffusion of CO\textsubscript{2}, transport and growth of the microalgae, and light irradiance can greatly facilitate the optimization of PBR design and operation\textsuperscript{16,17}. Given microalgae’s photosynthetic nature, these models require an accurate prediction of the overall light distribution throughout the PBR which is a function of reactor geometry, flow and mixing conditions, harvesting frequency, absorbance and scattering properties of the fluid and microalgae themselves.

Experimentally quantifying the light distribution within a bench-scale PBR, let alone a pilot or industrial scale PBR, is challenging using static sensors\textsuperscript{14,16}. This difficulty is apparent as published techniques for measuring light within experimental setups for cultivating microalgae vary widely among researchers\textsuperscript{18–22}. These difficulties in measuring light can make reproducible research extremely challenging and hamper efforts to quantify the efficiencies of PBRs. Experimentally measuring and validating the light exposure of microalgae using a moving reference frame within a PBR could provide valuable information about developing and optimizing PBR design and operation.
Objectives

This research focuses on developing a Lagrangian method for evaluating the light within PBRs using novel photochemical microsensors.

The objectives of the current research are to:

1. *Develop a chemically coated, microalgae-sized sensor that will undergo a change in fluorescence intensity when exposed to photosynthetically-relevant wavelengths of light.*

2. *Develop a method to quantify photochemical change in fluorescence on an individual microsensor basis using flow cytometry.*

3. *Evaluate the light distribution within a bench-scale, photosynthetic bioreactor using these microsensors.*
Past work with microsensors as non-biological surrogates

In recent decades, fluorescently-labeled microspheres, i.e. microsensors, have been used in conjunction with a flow cytometer to act as a non-biological surrogate to assess the effectiveness of varying methods of disinfection to inactivate bacterial pathogens such as Cryptosporidium\textsuperscript{23-25}. Past work has included the utilization of 1 µm, fluorescent microspheres to quantify the inactivation of Cryptosporidium by measuring the decay in fluorescence after sequential disinfection using both ozone and chlorine treatment\textsuperscript{10,11}. This decay in fluorescence that was caused by exposure to chlorine and ozone was used to characterize ‘survival ratios’ of Cryptosporidium oocytes. Fluorescently-labeled microspheres have also been used to evaluate the inactivation effectiveness of ultra violet (UV) reactors for drinking and wastewater treatment. In one instance, the decay in fluorescence, caused by photo-bleaching from UV light, has been used to make experimental measurements of the fluence distribution within the UV reactor and evaluate how these distribution measurements compared to those predicted with computational fluid dynamics\textsuperscript{23,25}. In a slightly different approach of Lagrangian actinometry, polystyrene microspheres were labeled with the caged fluorescent dye (E)-5-[2-(methoxycarbonyl)ethenyl] cytidine, which increases in fluorescence after UV light exposure, to quantify the UV dose distribution within a reactor\textsuperscript{26}. Clearly, the use of these dyed microspheres by researchers in drinking water disinfection has produced results that led to improved process performance assessments.
Microsensor development and relevant background

One of the main objectives of this research focuses on creating a non-biological surrogate for a microalgal particle that could be used to evaluate the light within PBRs. These microsensors will consist of microalgae-sized microspheres coated in a photo-reactive dye that changes in fluorescence when exposed to photosynthetic light; these changes in fluorescence will be quantified on a particle basis using flow cytometry (Figure 2.1).

**Figure 2.1** Microsensor overview diagram. A simplified overview of microsensor showing how 1) prior to light exposure, the photo-sensitive dye attached to microsphere will have little to no fluorescence when measured in a flow cytometer, 2) during light exposure, the dye reacts with light, uncaging fluorophore, and 3) after light exposure, the microsensor will increase in light intensity.
This sensor development process required investigation and an understanding of microspheres, photo-activated dyes, and flow cytometry measuring principles.

**Microsphere selection**

Microspheres are spherical particles that are often made from polystyrene plastic and range in size between 1-100 µm. Microspheres can be readily made with varying surface properties that allow for easy adsorption and permanent attachment of proteins and ligands and fluorescent labels\(^{27,28}\). For microspheres to mimic the movement of microalgal cells within a PBR, they must have the same basic physical properties. Therefore, the microspheres will need to have similar size and densities. Many polystyrene microspheres have density around 1.06 g·cm\(^{-3}\), slightly denser than water but close to neutrally buoyant like microalgae\(^{27}\). For the purposes of this research, two species of microalgae were considered when evaluating potential microspheres to act as surrogates: *Dunaliella* and *Nannochloropsis*. Both are marine microalgae that are considered potential organisms for biofuel production\(^{13,29-31}\). *Dunaliella* are rod-shaped, 6-10 µm single cell organisms that lack a cell wall\(^{32}\). The absence of a cell wall could reduce energy costs associated with lipid extraction. Although *Dunaliella* are around the same diameter as the microspheres, they are much more rod-shaped than the microspheres (Figure 2.2).
Moreover, *Dunaliella* have flagellar locomotive capabilities\(^{32}\), meaning that their flow characteristics might deviate more from the microspheres. *Nannochloropsis*, on the other hand, are non-motile, and spherical with a mean diameter between 2-4 µm. These physical characteristics, along with the advances made in genetically modifying *Nannochloropsis*\(^{33,34}\), make this microalgal species a strong candidate to model using the microspheres.

The surface properties of the microspheres are an important aspect of this potential microsensor because the surface groups are the substrate for the chemical attachment of the photo-reactive dye. The surface groups should not only be compatible with dye but, ideally, they should not cause aggregation among microspheres, which could change the flow properties and light exposure of the particles. The stability of the microspheres will likely depend on the conditions within the PBR in which these microspheres are tested such as solution pH, ionic strength, as well as the surface group (-NH\(_2\), -COOH, etc.), surface charge, number concentration, particle size\(^{27}\). Extensive aggregation can often be identified visually, but even aggregation below this visual threshold can cause issues. Other means of detecting aggregation include light microscopy (Figure 2.3), spectrophotometry, electrophoretic mobility, and field-flow fractionation\(^{27}\).
If aggregation is expected, or observed, polymer blockers such as polyethylene glycol (PEG) can be chemically attached to surface of the microspheres in the same manner as the dye to reduce aggregation potential\(^\textsuperscript{27}\). In the current research, microsphere-microalgae aggregation was not considered as these dyed-coated microspheres will be used to evaluate light within a PBRs without microalgae present.

**Photo-reactive dye selection**

There are two main types of photo-reactive mechanisms that were explored for this application: 1) Photo-bleaching and 2) Uncaging. In photo-bleaching, light irradiance slowly breaks down or degrades chemical bonds leading to a decrease in fluorescence intensity. The uncaging mechanism is based on the light breaking down a ‘cage’, i.e. a chemical bond, thereby allowing the fluorophore to fluoresce\(^\textsuperscript{35-37}\). Ideally, the photo-reactive dye would react to the same wavelengths of light utilized for microalgal growth (i.e. either in the blue and red regions of the spectrum: **Figure 2.4**).
Many of the photo-reactive dyes on the market are activated by UV light\textsuperscript{26,36}, as these wavelengths contain more energy than wavelengths near the infrared end of the spectrum. Thus, the activating wavelength for the proposed dye was targeted for the blue-end of the spectrum. After an extensive search for dyes activated in this visible range, one potential candidate was a caged-fluorescence dye produced by Abberior GmbH to meet the specifications described above. Abberior CAGE 552-N-hydroxysuccinimide ester (NHS-ester), one of the smallest and lightest caged rhodamine dyes available, is non-fluorescent before photolysis\textsuperscript{37}. After exposure to light in range of 360-440 nm, CAGE-552 NHS-ester transforms into a red fluorescent, photo-stable dye with a maximum absorption at 552 and maximum emission at 575 nm\textsuperscript{37}. This absorption and emission spectrums are suitable for fluorescence quantification using flow cytometry. Moreover, CAGE-552 NHS-ester is amine-reactive, meaning it can be readily coupled to the surface of microspheres modified with amine
groups\textsuperscript{27}, and, potentially coupled to other types of microspheres using chemical cross-linkers. Given that this caged dye is photo-activated by photosynthetic blue light, is capable of permanent surface attachment to microspheres, and has an absorption and fluorescent spectrum suitable for flow cytometry, it was considered the best candidate for this microsensor.

\textit{Measuring principle: flow cytometry overview}

In recent years, flow cytometry has become an increasing popular tool to evaluate and gain insight into individual, micron-sized particles\textsuperscript{38,39}. A flow cytometer has a unique fluidics system that consists of a central core, through which sample fluid flows, and an outer sheath fluid layer that is injected into the system at higher pressure. This difference in pressure results in a drag effect that forces sample particles into single-file line (Figure 2.5). The single-filed particles then pass through the laser excitation line, or series of lasers. As a particle passes through the laser beam, they create pulses of photons that are measured by optical detectors and converted into a voltage signal. These detectors have three ways of characterizing a pulse signal: height, area, and width. The signal height is defined as the maximum amount of current output (i.e. peak photon emission pulse) that occurs for a given particle, while the area signal is defined by the integral of the pulses over the time that the particle is in front of the laser. The pulse width is the total amount of time the particle spends in front of the laser\textsuperscript{40}.

Light that is scattered in the forward direction is collected into an optical detector known as the forward scatter channel (FSC); this channel is an indicator of size since larger particles typically refract more light than smaller ones. The side-scatter channel (SSC), placed 90 degrees to the laser excitation line, detects light scattered in this direction, providing valuable information about particle’s surface structure and granularity. Fluorescence detectors,
also placed 90 degrees from excitation line, have bandpass filters to capture specific wavelengths of light (Figure 2.5).

**Figure 2.5** Flow cytometry conceptual diagram. Illustration of internal components of flow cytometer showing hydrodynamic focusing and laser, optical detectors used for measuring individual particle characteristics.

Flow cytometer machines are configured with at least one excitation laser to measure forward and side scatter, but often have more than one laser with different excitation wavelength for measuring fluorescence. This research combines the aforementioned metrics to distinguish microsensor particles from other particles in the solution and to measure their
fluorescence intensity to estimate the cumulative light exposure received on an individual particle basis.
CHAPTER 3 - METHODS

Microsensor development

Abberior CAGE-552 dye, N-hydroxysuccinimide ester (NSH-ester), was used as the activating dye for this microsensor due to its unique ability to increase in fluorescence after exposure to photosynthetically-relevant blue light\textsuperscript{37}. 3 µm, amine-modified, polystyrene microspheres were chosen to represent \textit{Nannochloropsis} microalgae; a 2-4 µm spherical, non-motile, marine microalga. While negatively-charged carboxylated microspheres were initially preferred to the positively-charged amine microspheres (since the amine microspheres were more likely to aggregate with microalgae), the carboxylated microspheres were chemically incompatible for labeling with the CAGE-552 dye.

An organic synthesis reaction was used to couple to the amine-reactive dye to the amine surface-modified microspheres. Prior to this coupling reaction, the clean microsphere stock solution was vortexed for 2-4 minutes and then gently mixed with a rotator shaker – 10 rotations per minute (RPM) – at room temperature for around 20 minutes. The desired volume of microsphere stock was removed under a sterile hood (to prevent contamination of microsphere stock solution) and mixed with NaHCO\textsubscript{3} buffer (pH 8.3) in an appropriate-sized, light-resistant, glass vial. After calculating the amount of dye needed for coupling, the NHS-ester dye was dissolved into dimethyl sulfoxide (DMSO) solvent under a fume hood at a ratio of 50 µl DMSO to 1 mg NHS-ester. The dye-DMSO mixture was then added, drop-wise, to the buffer-microsphere solution as the solution was gently stirred; the coupling solution was then mixed for 1-2 hours at room temperature. After the coupling reaction was complete, the microspheres were washed by adding NaHCO\textsubscript{3} buffer to the solution at 2-4x the original
volume and vortexed for 1-2 minutes before centrifuging and decanting liquid. This washing procedure was repeated two to three times, using deionized (DI) water for the last washing cycle. After decanting the DI water from last wash cycle, 0.22 µm filtered DI water was added to bring the dyed microspheres to a desired stock concentration. The microsensors were stored at 4°C and used within two weeks for experimental purposes.

Preliminary tests showed visual evidence of aggregation for dye-coupled microspheres (Figure 3.1). Therefore, the aforementioned coupling procedure was modified to include the addition of a polyethylene glycol (PEG) polymer (m-dPEG®₂₄-NHS ester, Quantabiodesign®) to prevent aggregation.

![Figure 3.1](image)

**Figure 3.1** Visual evidence of aggregation of dyed microspheres. A) dyed-coated microspheres without polymer, B) dyed-coated microspheres with polymer, C) 0.22 µm filtered DI water as control for comparison.

The specifications for the microspheres, dye, and polymer that were used to calculate the appropriate experimental amounts for coupling are shown in Table 3-1, Table 3-2, Table 3-3. More detailed coupling information can be found in the Appendix (Table A1-1).
Table 3-1 Specifications for 3µm amine-surface labeled microspheres.

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diameter (µm)</td>
<td>2.90</td>
</tr>
<tr>
<td>Density at 20 C (g/ml)</td>
<td>1.06</td>
</tr>
<tr>
<td>Volume per microsphere (ml)</td>
<td>1.3E-11</td>
</tr>
<tr>
<td>Mass per microsphere (mg)</td>
<td>1.4E-08</td>
</tr>
<tr>
<td>Specific surface area (cm²/mg microsphere)</td>
<td>19.6</td>
</tr>
<tr>
<td>Parking area per NH₂ group (cm²)</td>
<td>2.1E-15</td>
</tr>
<tr>
<td>Micromoles NH₂ per microspheres (µMoles)</td>
<td>2.1E-10</td>
</tr>
<tr>
<td>NH₂ group per mg microsphere (#)</td>
<td>9.3E+15</td>
</tr>
<tr>
<td>NH₂ micromoles per gram microsphere (µMoles)</td>
<td>15.5</td>
</tr>
</tbody>
</table>

Table 3-2 Specifications for Abberior Cage-552 (NHS-Ester) caged dye.

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chemical name</td>
<td>C₃₀H₂₅N₅O₆ (NHS-ester)</td>
</tr>
<tr>
<td>Molecular weight (g/mol)</td>
<td>552</td>
</tr>
<tr>
<td>Molecules of dye per mg (#)</td>
<td>1.1E+18</td>
</tr>
<tr>
<td>NH₂ groups per molecule (#)</td>
<td>1.0</td>
</tr>
<tr>
<td>NH₂ groups per mg of dye (#)</td>
<td>1.1E+18</td>
</tr>
<tr>
<td>NH₂ molecules needed per microsphere (#)</td>
<td>1.3E+08</td>
</tr>
<tr>
<td>Dye needed per microsphere (mg)</td>
<td>1.2E-10</td>
</tr>
<tr>
<td>Safety factor for coupling rxn.</td>
<td>2.5</td>
</tr>
<tr>
<td>Dye needed per microspheres w. safety factor (mg)</td>
<td>2.9E-10</td>
</tr>
</tbody>
</table>

Table 3-3 Specifications for Quantabio design m-dPEG®2₄-NHS polymer.

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chemical name</td>
<td>m-dPEG®₂₄-NHS ester</td>
</tr>
<tr>
<td>Molecular weight (g/mol)</td>
<td>1214</td>
</tr>
<tr>
<td>Molecules of PEG per mg (#)</td>
<td>5.0E+17</td>
</tr>
<tr>
<td>NH₂ groups per molecule (#)</td>
<td>1.0</td>
</tr>
<tr>
<td>NH₂ groups per mg of PEG (#)</td>
<td>5.0E+17</td>
</tr>
<tr>
<td>NH₂ molecules needed per microsphere (#)</td>
<td>1.3E+08</td>
</tr>
<tr>
<td>PEG needed per microsphere (mg)</td>
<td>2.5E-10</td>
</tr>
<tr>
<td>Safety factor for coupling rxn.</td>
<td>2.5</td>
</tr>
<tr>
<td>PEG needed per microsphere w. safety factor (mg)</td>
<td>6.3E-10</td>
</tr>
</tbody>
</table>
**Quantifying dosage-response kinetics of microsensor**

**Collimated beam tests and setup**

A collimated beam apparatus was set up to expose microsensor samples to a known, uniform, collimated light source. This enclosed, box-like setup was constructed based on the protocol of Bolton et al. (2003) as shown in Figure 3.2.

![Diagram of collimated beam apparatus](image)

**Figure 3.2** Diagram of collimated beam apparatus used to expose microsensor samples to uniform light source.

A blue LED grow lamp with a peak intensity around 440 nm (Appendix, Figure A1-1) was used as the activating light source. For each dosage test, a 20 mL sample of dyed microspheres and 0.22 µm filtered DI water (number concentration: 40-200,000/mL) were placed in a glass petri dish that was centered within the collimated setup. The light intensity across the petri dish was measured using a USB Ocean Optics Spectrometer at 0.5 cm increments with a specialized 3-D printed sensor holder (Figure 3.3).
A petri-factor, as described by Bolton\textsuperscript{41}, was calculated from this data to estimate the uniformity of the light source across the surface based on a scale from 0-1 with 1 being perfectly uniform. This factor was maintained between 0.9-1.0 to achieve appropriate uniformity of the surface light irradiance on the petri dish.

**Sample analysis in flow cytometer**

A Beckman Coulter CytoFLEX © flow cytometer configured with two lasers (488 and 633 nm) was used to quantify microsensor fluorescence using a 585/40 nm filtered fluorescent channel. This filter best fit the emission range of uncaged fluorescent dye (Appendix, Figure
A1-2). Before analyzing samples, the machine was prepared by running a cleaning solution for 10-15 minutes and subsequently performing quality control as specified by manufacturer. All samples were vortexed for 1 minute before analysis and the sample lines were flushed with cleaning solution and DI water for 1 minute between sample runs.

Gating is a flow cytometry technique used for selecting populations based on shared characteristics and is commonly used to isolate regions of interest for data analysis. Since the microspheres will ultimately act as a non-biological surrogate for microalgal particles, selecting the correct microspheres (i.e. the microspheres intended to represent microalgae) from all flow cytometry events is very important. While in theory, the hydrodynamic focusing within the flow cytometer should lead to only individual particle entering the laser chamber at a time, particle aggregates are possible. This particle-particle interaction is likely if the particles and conditions are favorable for aggregation. Two different gates were used to select only single-microsphere populations within the targeted 3 µm size range (Figure 3.4). The first gate was based on the relative particle size, while the second was based on the relative granularity, or scatter, of the particle. Aggregated particles are often identified by examining the pulse area with respect to the pulse height, or width, of a channel, particularly the FSC channel. Visual deviations observed when examining these relationships are often considered doublets, or two particles flowing past the laser at the same time.
Figure 3.4 Flow cytometry density plots showing gating technique used to select consistent, non-aggregated microsphere populations. These gates were chosen based on A) FSC-H vs FSC-A, or relative particle size height vs area, Gate #1, and B) SSC-H vs SSC-A or relative particle scatter/granularity height vs area, Gate #2.
FSC Express 5 Flow Research Edition© software was used to define gates and to export microsphere data meeting both criteria as a common delimited files. Selecting the correct gate ensures population distributions remain consistent with respect to the number (single vs. double event), size, shape, granularity, and the fluorescence of the particles over time and ranging experimental conditions. For all experiments, 2-3 technical replicates were analyzed for each collimated light exposure test, each with 5,000-10,000 microsphere events within the targeted gate range.

**Signal processing microsensor fluorescence data**

Preliminary dose-response tests confirmed a similar response in microsphere fluorescence as previously reported when using UV-reactive microsensors\textsuperscript{25,26}, e.g. non-normal, overlapping fluorescence distributions across varying dosages (Figure 3.5).

![Figure 3.5](image)

**Figure 3.5** Fluorescence of microsensors over range of dosages. These kernel density estimates of the fluorescence show overlapping fluorescence values and increasing variance with increasing dosage.
This overlap and non-uniform shift is likely due to a number of factors including 1) physical variations in the particle (e.g., size, surface granularity), 2) variations in dye attachment concentration (i.e. the number of dye molecules attached on any given microsphere surface), 3) the uncaging mechanisms (the probability of the light source ‘striking’ a caged fluorophore decreases as more fluorophores are uncaged), and 4) the ‘point-spread function’ of the machine (i.e. how the machine’s optical detectors responds to and processes a point-source of light). Regardless of the cause, this spread in fluorescence over varying dosages can create a challenge for predicting the light exposure as a direct function of fluorescence.

**Deconvolution**

As described in Blatchley\textsuperscript{26}, deconvolution can be used to extract the underlying dosage distribution from a sample’s fluorescence distribution obtained from the flow cytometer. Deconvolution is a signal processing technique commonly used in scientific and engineering\textsuperscript{42,43} applications to recover a desired signal, \( \mathbf{y} \), from a recorded signal, \( \mathbf{d} \), that has been convolved by a transfer function, \( \mathbf{a} \), (Equation 3.1).

\[
d = \mathbf{y} \ast \mathbf{a}
\]

(Equation 3.1)

This algorithm-based process is easily represented in matrix form as described in Equation 3.2 with constraints specified in Equations 3.3-3.5.
\[
\sum_{i=0}^{m-1} \gamma_i = 1 \tag{3.3}
\]
\[
\sum_{j=0}^{n-1} \delta_j = 1 \tag{3.4}
\]
\[
\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \alpha_{i,j} = m \tag{3.5}
\]

where,

\( q = \text{fluorescence bin width}, \)
\( p = \text{dose bin width}, \)
\( n = \text{number of fluorescence bins}, \)
\( m = \text{number of dose curves}, \)
\( \alpha_{i,j} = \text{Fraction of beads receiving Dose}_{pj}, \text{expected to fluoresce at } FI_{qi}, \)
\( \gamma_j = \text{Fraction of all beads in convolved sample that received dose Dose}_{pq}, \)
\( \delta_i = \text{Fraction of all beads in convolved sample that fluoresce at } FI_{qi}. \)

Each column in matrix \( A \) represents a fluorescence distribution density curve generated from a collimated beam sample; the rows represent fluorescence intensity bins. In other words, \( \alpha_{ij} \) represents the fraction of all microspheres that receive a collimated \( \text{Dose}_{pq} \) that are expected to fluoresce within fluorescence bin \( FI_{qi} \). Given \( A \) and \( \Gamma \), vector \( D \) can be estimated using matrix multiplication, \( D^* = A \times \Gamma \). Thus, given both \( A \) and \( D \), \( \Gamma \) can be predicted, \( \Gamma^* \), by minimizing the difference between \( D \) and \( D^* \), while enforcing the constraint shown in Equation 3.3. Equations 3.4 and 3.5 are both satisfied by using kernel density functions generated from the flow cytometry output as matrix \( A \) and vector \( D \) (Appendix, Section 2-1). The difference between the predicted and actual fluorescence distribution of the convolved sample was measured by the residual square error (RSE) vector (Equation 3.6).

\[
\text{RSE} = (A \Gamma^* - D)^2 \tag{3.6}
\]
The sum of this RSE vector is the residual sum of squares (RSS). The deconvolution approach was implemented in Python 3.5.1 using the Minimize function within Python’s Non-Linear Least-Square Minimization and Curve-Fitting package (lmfit). A built-in, modified Powell’s method was used as the optimizing method within this function with a convergence tolerance of 0.0001 and maximum number of function evaluations equal to $p \times 1000$, where $p$ is the number parameters (or in this case, the number of light dosage bins, $m$). This modified Powell’s method is a conjugate direction method, performing sequential one-dimensional minimizations along the RSE vector; derivatives are not needed for this method. For each convolved sample, the deconvolution optimization function was called 100 times, each time with random initial dose estimates. The full algorithm and associated scripts are presented in the Appendix, Section 3.

Deconvolution with interpolation

The resolution of the dose distribution output from deconvolution will depend on the number fluorescence distribution standards used within the algorithm, (Figure 3.5, Equation 3.2). Microalgae within a PBR will likely experience a wide range of light dosages but it is infeasible to experimentally determine a fluorescence distribution curve for each one of these dosages. If, however, the fluorescence distribution of the microsensors can be predicted as a function of the light exposure dosage, then these distribution curves can be interpolated between experimental light dosages as shown in Figure 3.6. The first step to improving deconvolution with interpolation is to mathematically characterize how these distributions change with increasing light exposure.
Figure 3.6 Example of interpolated curves describing the distribution of microsensor fluorescence for different exposures to a collimated light source.

Preliminary results indicated that the fluorescence distribution of microspheres follow a stable distribution; a family of stable distributions are infinitely divisible and generally heavy-tailed distributions. More specifically, the stable distribution is defined by four parameters: Stability, $\alpha \in (0, 2]$; Skewness, $\beta \in [-1, 1]$; Scale, $c \in (0, \infty)$; and Location, $\mu \in (-\infty, \infty)$. The stable distribution can be described using the Fourier transform of its characteristic functions as shown in Equations 3.7-3.10.

\[
f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \varphi(t) e^{-ixt} dt
\]

\[
\varphi(t, \alpha, \beta, c, \mu) = \exp[it\mu - |ct|^\alpha (1 - i\beta \text{sgn}(t) \Phi)]
\]

\[
\Phi = \tan\left(\frac{\pi\alpha}{2}\right), \alpha \neq 1
\]

\[
\Phi = -\frac{2}{\pi} \log|t|, \alpha = 1
\]
MATLAB’s `AllFitDist` function was used to characterize the fluorescence distribution over the light dose range tested and to determine the best distribution fit for the data. This package fits data to multiple parametric probability distributions and sorts these fits using multiple criteria, including Bayesian information criterion (BIC) which was used as the primary sorting criteria for distribution selection; a list of distributions tested using MATLAB is provided in the Appendix (Section 2-2). The BIC is a criterion for model selection that considers the residual sum of squares (RSS) of n data points while also penalizing for the number of model parameters (p) to reduce overfitting. When comparing multiple models to describe one data set, the model with lowest BIC is considered superior. The BIC value was calculated according to Equation 3.11:

\[
\text{BIC} = n + n \log(2\pi) + n \log \left( \frac{\text{RSS}}{n} \right) + (\log n) (p + 1)
\]

### Dosage-response kinetics experimental procedure

A range of light dosages up to 20 kJ-m⁻² were analyzed using the collimated beam setup and flow cytometer. These collimated beam tests were performed at two different light intensity levels that may occur in the PBR before inoculating the reactor with microalgae. Exposure times for collimated beam experiments ranged from 1 to 25 minutes. Longer exposure times were avoided to prevent any impact of heating from the grow lamp on the dye kinetics. Table 3-1 display the experimental conditions for the collimated beam tests.

<table>
<thead>
<tr>
<th>Set #</th>
<th>Light Intensity (µW-cm⁻²)</th>
<th>Exposure Time (min)</th>
<th>Dosage (kJ-m⁻²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set 1</td>
<td>1863</td>
<td>1, 6, 12, 18</td>
<td>1, 7, 13, 20</td>
</tr>
<tr>
<td>Set 2</td>
<td>1334</td>
<td>1.5, 12, 12, 25</td>
<td>1, 10, 10, 20</td>
</tr>
</tbody>
</table>
The purpose of these dose-response tests was to 1) verify that the fluorescence distribution of microsensors across this dose range was best described stable distribution, 2) determine the dose-response kinetics of microsensors with respect to the four stable distribution parameters between 1-20 kJ-m\(^2\) and 3) verify reciprocity in response for the two intensities tested.

**Proof-of-concept testing**

*Predicting a known light distribution with deconvolution*

To test the deconvolution technique, ‘convolved’ samples were prepared by mixing microsensor samples that were exposed to known, collimated light dosages. The output of the fluorescence distribution for these convolved samples were analyzed via the deconvolution algorithm. The fluorescence input curves used for deconvolution were generated from the constituent collimated beam samples that went into each sample. Deconvolution with interpolation was not used. Four different samples of statistical distributions were tested: even, unimodal, biomodal, and skewed. The known light exposure dose distribution of each sample was calculated using the concentration of each dose, which was measured by the flow cytometer, and the volume added to the mixture sample. A simplified visualization of the convolved sample preparation for a bimodal distribution is shown in Figure 3.7.
Figure 3.7 Convolved sample preparation diagram showing how a bimodal distribution sample would be prepared using 5 distinct dosages if the microsensor number concentration were assumed constant for all five dose components.

**Predicting the light distribution within a bench-scale PBR**

A 3 L batch reactor was designed as part of ongoing research to determine microalgal growth kinetics. This 8.25 x 3.5 x 8.375-inch reactor was constructed from 0.44 inch optically clear, cast acrylic plastic. The mixer is slightly off centered to allow for a light sensor to be centered within the reactor (Figure 3.8).

Figure 3.8 Bench-scale PBR used for proof-of-concept testing. A) reactor as setup for proof-of-concept tests, and B) schematic drawing given in inches.
Conditions for this proof-of-concept test were set up to replicate kinetic batch tests with sensors, tubing, and diffuser within the reactor, and an axial impeller mixing at ~60-80 RPM.

Both the impeller and the mixing intensity were chosen to minimize shear that can harm microalgae. The characteristic mixing time, $\tau$, of this reactor configuration was determined to be around 2 minutes using a simple dye test (Figure 3.9); the characteristic mixing time is assumed to represent the average amount of time required for a particle or fluid parcel to travel...
around the whole reactor and was used to set the exposure times for sampling. In theory, if the average light dose and distribution is known for this characteristic mixing time, despite the time scale being much less than that for microalgal growth, light exposure estimates for longer time scales (considered multiple replications of the characteristics time scales) can be estimated assuming there is no change in fluid flow or light transmittance over longer time scales.

The microsensors were added to the reactor after filling it with 3.0 L of DI water. The PBR was then mixed for 1.5 $\tau$ prior to turning on the light. 10 mL samples were removed at 1, 2, 4, and 8 minutes of light exposure ($\frac{1}{2}\tau$, $\tau$, $2\tau$, and $4\tau$). At these time increments, the light was turned off and the PBR continued mixing for 1.5 $\tau$ to ensure that the sampled volumes were representative of the reactor as a whole. Samples were removed from both the right and left side at approximately the same depth.

Collimated beam tests were performed immediately prior to these reactor tests at light dosages of 1.12, 6.71, 13.41, and 20.1 kJ-m$^{-2}$ (Table 3-4, Set 2). These light dosages represent a range of expected values based on starting light conditions for algal kinetic tests. The fluorescence output from these collimated samples were fit to a number of distributions (Appendix, Section 2-2). The relationships between the parameters of the best fitted distribution, as defined by $BIC$, and light dosages were then established using the experimental collimated beam data. Using these relationships, fluorescence distributions curves were interpolated between 1 to 20 kJ-m$^{-2}$ at a dosage increments of 0.25, 0.5, 1 kJ-m$^{-2}$. These dosage bin widths were chosen to explore if, and how, the predicted light dose distribution change as more dosage curves are input (i.e., smaller dose increments) into the deconvolution algorithm and how binning impacts the model strength with respect to $BIC$ and $RSS$. 
A handheld spectrometer was also used to measure the light intensity at positions across the reactor body to 1) center the light, 2) ensure that the light dosage range over the course of the experiment would fall between the dosages calibrated with the collimated beam tests, and 3) estimate an average light intensity for PBR. Due to limitations with the spectrometer, measurements were not taken inside the reactor or within the fluid, and simplifications were made for estimating the average light intensity. This average light intensity was estimated by discretizing the reactor into three sections as shown in Figure 3.10.

**Figure 3.10** Positions across PBR at which light intensity measurements were taken with a handheld, visible light spectrometer. A, B, C, F, and G were used to center the PBR body. B, D, E were used to estimate the average light intensity in the PBR; these values were assumed to be representative of the average light within components #1, #2, #3 respectively.
Three central measurements were taken across the reactor depth and these measurements were used to represent the average light intensity for each section. These intensities were then used to compute an average intensity measurement estimate for the entire PBR. As a further validation tool, collimated samples were analyzed with the deconvolution algorithm in the same manner as the PBR samples to gain insight into how well this deconvolution method predicts both the average dose and dose distribution of these collimated samples.
CHAPTER 4 - RESULTS & DISCUSSION

Quantifying microsensor dosage-response behavior and kinetics

Distribution fitting

In all samples exposed to a collimated light dosage, the stable distribution, followed closely by both the 3-parameter, t-location scale distribution and 2-parameter, log-logistic distributions, provided the best fit in describing the microsensor fluorescence among all the distributions considered based on the BIC criteria (Figure 4.1).

Figure 4.1 BIC criteria values for most promising distributions for describing the distribution of the microsensor fluorescence after exposure to known, uniform light dosage. Values represent average BIC across all samples replicates and dosages. Note: Distribution fitting analysis was done for each sample, across all dosages and replicates; the stable distribution consistently had lowest BIC.

Figure 4.2 displays the graphical representation of the numerical stable distribution fit compared to the empirical data across the dosage range tested.
After verifying that the microsensor fluorescence data followed a stable distribution under collimated light conditions, the parameters that define this distribution (α, β, c, μ) were analyzed with respect to the applied light dosage (Figure 4.3).

**Figure 4.2** Empirical versus stable fitted fluorescence distribution for microsensors over range of light dosages experimentally measured in collimated beam apparatus.

**Dose-response parametric equations**

After verifying that the microsensor fluorescence data followed a stable distribution under collimated light conditions, the parameters that define this distribution (α, β, c, μ) were analyzed with respect to the applied light dosage (Figure 4.3).
Figure 4.3 Stable distribution parameters versus light dosage. A) $\alpha$ -stability, B) $\beta$ -skewness, C) $\epsilon$ -scale, and D) $\mu$-location parameter versus light dosage under a 1330 $\mu$watt-cm$^{-2}$ (PINK) and 1880 $\mu$watt-cm$^{-2}$ (GREEN) collimated light intensity.

Linear, quadratic, and cubic models were evaluated for each parameter using the $BIC$ criteria (Table 4-1). The stability and scale parameters appeared to follow a quadratic model, while the skewness and location parameters were better represented with a cubic function. This assessment was made by evaluating the $BIC$ of a linear, quadratic, and cubic model.

Table 4-1 BIC criteria values used to evaluate best fit for stable parameter curves.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Linear</th>
<th>Quadratic</th>
<th>Cubic</th>
<th>Best</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stability</td>
<td>-94</td>
<td>-117</td>
<td>-117</td>
<td>Quadratic</td>
</tr>
<tr>
<td>Skewness</td>
<td>9</td>
<td>12</td>
<td>3</td>
<td>Cubic</td>
</tr>
<tr>
<td>Scale</td>
<td>386</td>
<td>382</td>
<td>465</td>
<td>Quadratic</td>
</tr>
<tr>
<td>Location</td>
<td>503</td>
<td>500</td>
<td>465</td>
<td>Cubic</td>
</tr>
</tbody>
</table>
For the experimental dosages tested, the microsensors did not appear to reach saturation (i.e. fully uncage and reach a maximum fluorescent intensity). The saturation point of these microsensors is an important consideration and a test that should be performed to assess their full range and to ensure best better fitting techniques when establishing relationships between distribution parameters and dosage values for deconvolution. For application purposes, these dosage-response relationships should be determined each time within the expected light dosage range before analyzing samples containing an unknown light dose distribution. Moreover, these different fitting functions should not be used to predict outside the applied light dosage range used for calibration.

*Reciprocity*

For the light dose range and intensities tested, the uncaging reaction of the microsensors with respect to light dose, as described by stable parameters, did not appear to change with light intensity level (Figure 4.3), suggesting that reciprocity holds for two light intensity values tested. However, since only two light intensities levels were tested, and, for those two levels, a limited number of exposure times, i.e. light dosages, were tested (n=3, n=4), more dosage replicates and intensity levels should be tested to statistically conclude that reciprocity holds for the uncaging reaction of these microsensors.

*Proof-of-concept testing*

*Predicting a known light dosage distribution*

As previously stated, deconvolution works to resolve a light dosage distribution by minimizing the difference between a predicted fluorescence distribution and the measured
signal output from flow cytometer. For the four convolved samples tested, the deconvolution algorithm closely predicted the measured fluorescence distribution (Figure 4.4).

**Figure 4.4** Fluorescence signal output as measured by flow cytometer and shown as a kernel density function (BLUE) as compared to signal output estimated with deconvolution algorithm (ORANGE) for convolved samples with A) even, B) unimodal, C) bimodal, and D) skewed dosage distributions.
The estimated light dosage distributions that generated the predicted fluorescence output shown in **Figure 4.4** also produced good estimates for the expected light dosage distribution of the convolved samples, which was calculated based on volume and concentrations measurements of constituent sample parts (**Figure 4.5**).
Figure 4.5 Measured vs deconvolution predictions for light dosage distribution for convolved samples. Estimates predicted based on concentration and volume (LEFT) and deconvolution (RIGHT) for A) even, B) unimodal, C) bimodal, and D) skewed dosage distributions. Error bars represent standard error for n = 3 samples.
**Predicting the light distribution of bench-scale PBR**

The fluorescence distribution for PBR samples after different exposure times within the PBR is shown in Figure 4.6. These curves appeared to mimic fluorescence distribution of microsensors after exposure to collimated light source (Figure 4.2), which is not unexpected as this PBR was designed to keep the light intensity as uniform as possible throughout the reactor.

![Fluorescence distribution curves](image)

**Figure 4.6** Fluorescence distribution curves from experimental samples after 1, 2, 4, and 8 minutes of light exposure within proof-of-concept PBR. Fluorescence distributions are shown as smoothed kernel density functions based on flow cytometry fluorescence output.

Before using deconvolution to predict the light dosage distribution based on the fluorescence output as shown in Figure 4.6, the light dosage relationships for stable parameters previously established (Figure 4.3) were confirmed for the collimated beam tests conducted prior to testing microsensors within the PBR. Fluorescence distributions curves across
expected dosage range were then interpolated using these relationships at light dose increments of 0.25, 0.5, and 1 kJ-m⁻². **Figure 4.7** shows the interpolated curves for a dose width of 1 kJ-m⁻² and how these compare to the experimentally determined fluorescence distributions.

![Figure 4.7](image-url)

**Figure 4.7** Predicted fluorescence distribution curves generated by interpolation of stable distribution parameters compared to fluorescence distribution curves generated experimentally via collimated beam testing. Interpolated curves are shown for light dosages between 1 and 20 kJ-m⁻² at a light dosage bin width of 1 kJ-m⁻²

For each bin width tested, the generated interpolated distribution curves were input into the deconvolution algorithm, along with the measured fluorescence distribution output from the reactor samples, to predict the light dose distribution within the reactor. The light distributions predictions for all PBR samples are shown in **Figure 4.8**, along with how these predictions vary by changing the bin width of dosage curves used for deconvolution.
Figure 4.8 Average light dose distribution predictions for PBR at 1, 2, 4, and 8 minutes of light exposure using an interpolated light dosage bin width of A) 1, B) 0.5, and C) 0.25 kJ-m⁻². Error bars represent one standard deviation in average dosage prediction after n = 100 optimization trials.
Validation and impact of light dosage bin width on light estimates

To compare the validity of these PBR light dosage predictions, an average light dosage was made by estimating the average light intensity within PBR based on measurements taken across the width of the PBR with the spectrometer (Table 4-2).

Table 4-2 Light intensity measurements taken across PBR for proof-of-concept testing. Light measurements across the reactor depth, without water, taken with handheld spectrometer. Note: Letter references refer to Figure 3.11.

<table>
<thead>
<tr>
<th>Position</th>
<th>Light Intensity (µW·cm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Front (B)</td>
<td>2870</td>
</tr>
<tr>
<td>Middle (D)</td>
<td>1383</td>
</tr>
<tr>
<td>Back (E)</td>
<td>749</td>
</tr>
<tr>
<td><strong>Average:</strong></td>
<td><strong>1667</strong></td>
</tr>
</tbody>
</table>

The average predicted dosage was estimated as the product of the intensity and the total time the microsensors were exposed to light within the PBR. The estimated dosage was then compared to predicted average light dosage from microsensors using deconvolution. The accuracy of the microsensor/deconvolution method was also evaluated by analyzing the average light dosage of three collimated samples via deconvolution. These comparisons are shown in Table 4-3.
Table 4-3 Deconvolution prediction for the average light dosage from the PBR and collimated samples as compared to measured average light dosage estimates. Deconvolution predictions made with dosage bin widths of 1, 0.5, 0.25 kJ-m-2. *Indicates that the deconvolution predictions for the collimated samples fall within a 95% confidence interval of the spectrometer-based measurements. Note: Confidence intervals for the measured average light dosage in PBR were not computed due to measurement limitations.

<table>
<thead>
<tr>
<th>Bin Width</th>
<th>Predicted avg. light dosage w. deconvolution (kJ-m-2)</th>
<th>Estimated avg. light dosage w. static sensor (kJ-m-2)</th>
<th>Confidence interval for estimated avg. with static sensor (95%)</th>
<th>% difference</th>
<th>Optimized RSS value for deconvolution</th>
<th>BIC value for deconvolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 min PBR</td>
<td>1.51</td>
<td>1.00</td>
<td>--</td>
<td>51%</td>
<td>2.77E-09</td>
<td>-3060</td>
</tr>
<tr>
<td>2 min PBR</td>
<td>3.08</td>
<td>2.00</td>
<td>--</td>
<td>54%</td>
<td>5.90E-10</td>
<td>-3285</td>
</tr>
<tr>
<td>4 min PBR</td>
<td>5.38</td>
<td>4.00</td>
<td>--</td>
<td>35%</td>
<td>3.99E-09</td>
<td>-3007</td>
</tr>
<tr>
<td>8 min PBR</td>
<td>8.91</td>
<td>8.00</td>
<td>--</td>
<td>11%</td>
<td>3.72E-10</td>
<td>-3353</td>
</tr>
<tr>
<td>1.1 kJ-m-2</td>
<td>1.27</td>
<td>1.12</td>
<td>[1.02-1.21]</td>
<td>13%</td>
<td>6.19E-09</td>
<td>-2943</td>
</tr>
<tr>
<td>6.7 kJ-m-2</td>
<td>6.72*</td>
<td>6.71</td>
<td>[6.61-6.80]</td>
<td>0%</td>
<td>3.57E-09</td>
<td>-3023</td>
</tr>
<tr>
<td>13.4 kJ-m-2</td>
<td>13.41*</td>
<td>13.41</td>
<td>[13.30-13.52]</td>
<td>0%</td>
<td>2.48E-10</td>
<td>-3415</td>
</tr>
<tr>
<td>1 min PBR</td>
<td>1.50</td>
<td>1.00</td>
<td>--</td>
<td>50%</td>
<td>2.76E-09</td>
<td>-3023</td>
</tr>
<tr>
<td>2 min PBR</td>
<td>3.07</td>
<td>2.00</td>
<td>--</td>
<td>54%</td>
<td>6.13E-10</td>
<td>-3241</td>
</tr>
<tr>
<td>4 min PBR</td>
<td>5.41</td>
<td>4.00</td>
<td>--</td>
<td>35%</td>
<td>8.66E-10</td>
<td>-3191</td>
</tr>
<tr>
<td>8 min PBR</td>
<td>8.92</td>
<td>8.00</td>
<td>--</td>
<td>12%</td>
<td>7.16E-10</td>
<td>-3220</td>
</tr>
<tr>
<td>1.1 kJ-m-2</td>
<td>1.26</td>
<td>1.12</td>
<td>[1.02-1.21]</td>
<td>13%</td>
<td>6.13E-09</td>
<td>-2907</td>
</tr>
<tr>
<td>6.7 kJ-m-2</td>
<td>6.69*</td>
<td>6.71</td>
<td>[6.61-6.80]</td>
<td>0%</td>
<td>1.20E-09</td>
<td>-3143</td>
</tr>
<tr>
<td>13.4 kJ-m-2</td>
<td>13.42*</td>
<td>13.41</td>
<td>[13.30-13.52]</td>
<td>0%</td>
<td>1.88E-10</td>
<td>-3415</td>
</tr>
<tr>
<td>1 min PBR</td>
<td>1.54</td>
<td>1.00</td>
<td>--</td>
<td>54%</td>
<td>2.69E-09</td>
<td>-2951</td>
</tr>
<tr>
<td>2 min PBR</td>
<td>3.08</td>
<td>2.00</td>
<td>--</td>
<td>54%</td>
<td>6.98E-10</td>
<td>-3147</td>
</tr>
<tr>
<td>4 min PBR</td>
<td>5.40</td>
<td>4.00</td>
<td>--</td>
<td>35%</td>
<td>8.11E-10</td>
<td>-3125</td>
</tr>
<tr>
<td>8 min PBR</td>
<td>8.92</td>
<td>8.00</td>
<td>--</td>
<td>12%</td>
<td>4.95E-10</td>
<td>-3201</td>
</tr>
<tr>
<td>1.12 kJ-m-2</td>
<td>1.43</td>
<td>1.12</td>
<td>[1.02-1.21]</td>
<td>28%</td>
<td>5.87E-09</td>
<td>-2838</td>
</tr>
<tr>
<td>6.71 kJ-m-2</td>
<td>6.68*</td>
<td>6.71</td>
<td>[6.61-6.80]</td>
<td>0%</td>
<td>8.96E-10</td>
<td>-3110</td>
</tr>
</tbody>
</table>

The greatest percent difference between the spectrometer-based and microsensor-based dosage prediction for the PBR was around 50-55%, which occurred at the 1 and 2-minute exposure times. This difference in estimates could be due to a number of factors, such as the transmittance and reflection from PBR wall and water layer that were not considered here.
However, as also noted in Table 4-3, the deconvolution prediction for average light dosage for the 1.12 kJ-m\(^{-2}\) collimated sample did not fall within the 95% confidence interval of measured prediction like the predictions for the 6.71 and 13.41 kJ-m\(^{-2}\) collimated samples. The light dose distribution predictions for the collimated samples (Figure 4.9) provide insight into why the prediction error for samples exposed to lower light dosages are greater for both the PBR samples and collimated light samples. As noted in Figure 4.9 (and also in Figure 4.8), the deconvolution algorithm consistently—over all bin widths—attempts to include light dosage components from around the 4-6 kJ-m\(^{-2}\) range into the distribution predictions for the microsensors exposed to 1.12 kJ-m\(^{-2}\) of collimated light.
Figure 4.9. Light dosage predictions for samples exposed to a known, collimated light source using deconvolution with an interpolated dosage bin width of A) 1, B) 0.5, and C) 0.25 kJ-m⁻². Error bars represent the standard deviation of solutions after n = 100 optimization trials.

The inclusion of these dosage components that fall outside range of expected values is likely caused by the distribution fitting/interpolation error, i.e. the stable curve predictions are not adequately describing the microsensor fluorescence distributions for low light dosages. Thus, when the algorithm attempts to re-create the experimental fluorescence distributions for
the 1.12 kJ-m\(^2\) sample, a smaller error (RSS) can occur by incorporating small components from other dosages as shown in Figure 4.10.

**Figure 4.10** Graphical visualization of the impacts of fitting/interpolation error on deconvolution results when analyzing the 1.12 kJ-m\(^2\) collimated sample. Deconvolution attempts to reconstruct, as shown in RED, the fluorescence distribution of the 1.12 kJ-m\(^2\) experimental sample (PURPLE) using the interpolated curves (BLUE). For the 1.12 kJ-m\(^2\) sample, the deconvolution algorithm can achieve a smaller minimum – the RSS between the reconstructed (RED) and the actual (PURPLE) fluorescence distributions – by assuming a small fraction of the experimental sample was exposed to light dosages in the 4-6 kJ-m\(^2\) range. The predicted dosage distribution is shown as percentages next to interpolated curves.

The impact of this fitting error observed on the distribution predictions highlights another important consideration when utilizing this deconvolution method for predicting light distributions: overfitting. **Figure 4.11** illustrates this overfitting problem by showing how the percent difference from the measured average light dosage and the RSS change with binning for the 1.12 kJ-m\(^2\) collimated sample. One will note that while the RSS value decreases when
decreasing the bin width from 1 kJ-m\(^2\) to 0.25 kJ-m\(^2\), the overall percent difference in the predicted dosage increases. The \(BIC\) values indicate that the deconvolution solution with a bin width of 1 kJ-m\(^2\) is the best model for predicting the light distribution of this sample (Table 4-3).

**Figure 4.11** Residual sum of square (ORANGE) and the percent difference between spectrometer-based estimates for average light dosage and deconvolution predictions (BLUE) at three light dosage bin widths (0.25, 0.5, 1.0 kJ-m\(^2\)). This graphic highlights one implication of overfitting in deconvolution, i.e. decreasing the bin width can lead to an overall lower RSS for optimization but a higher error from true value.

**Figure 4.12** illustrates how the light dosage bin width impacts deconvolution’s ability to reproduce the recorded fluorescence signal, as indicated by the \(RSS\) and \(BIC\), for the PBR samples tested. The extent of this impact was sample dependent, as clearly demonstrated by the stark increase in \(RSS\) when the dosage bin width was increased from 0.5 to 1 kJ-m\(^2\) for the 4-minute sample; such a drastic change in \(RSS/BIC\) with changing bin width was not observed in any other PBR sample. Note, the *average* light dosage prediction for this 4-minute sample
was consistently around 5.4 kJ-m\(^2\) across all bin widths (Table 4-3). Depending on what type of distribution produced this averaged value, the deconvolution algorithm may struggle to reproduce the recorded fluorescence using only dosage curves at 1 kJ-m\(^2\) increments; in other words, when using a bin width of 1 kJ-m\(^2\), the algorithm must assume that around half the microsensors within the sample were exposed to a light dose of 5 kJ-m\(^2\) and the other half to a light dose of 6 kJ-m\(^2\), when in reality, the majority of the microsensors could have been exposed to a single light dosage closer to 5.5 kJ-m\(^2\). In agreement with the latter assertion, the \textit{BIC} value for this sample suggests that the deconvolution solution produced using a dosage bin width of 0.5 kJ-m\(^2\) most adequately describe the true distribution of this 4-minute sample without overfitting.

\textbf{Figure 4.12} BIC (LINE) and RSS (COLUMN) values for dosage distribution predictions for samples after 1, 2, 4, and 8 minutes of light exposure within PBR. Predictions were made using deconvolution where bin width for interpolated fluorescence distribution curves were varied from 1 (ORANGE), 0.5 (BLUE), and 0.25 (GREEN) kJ-m\(^2\).
These binning observations suggest a tradeoff: in some cases, decreasing the bin width could give a more accurate description of the true distribution (*example*: 4-minute PBR sample, 1 vs 0.5 kJ-m⁻²), while in other scenarios, this decrease could lead to overfitting and an overall less accurate prediction of both the average light intensity and the light distribution (*example*: 1.12 kJ-m⁻² collimated sample, 1 vs 0.25 kJ-m⁻²). While a more detailed analysis of binning is needed to fully understand its implications on this method, the few observations performed in this study suggest a better way to implement this deconvolution algorithm for predicting light distributions within a PBR. This new approach consists of including both the dosage bin width as a parameter within the optimization structure and minimizing the BIC instead of RSS (as opposed to manually changing the dosage bin width as an input and examining its impacts on BIC as done in this work).
CHAPTER 5 - CONSIDERATIONS, FUTURE WORK, & CONCLUSIONS

Deconvolution and light dosage distribution estimates

A more thorough analysis of the deconvolution algorithm with interpolation should be considered to better assess (and improve) this method’s ability to predict the light dosage distribution of an unknown sample. Some factors that could impact these predictions include, but are not limited to, 1) the numerical distribution fit describing the fluorescence distribution after exposure to a collimated light source, 2) the range and size of the light dosage bins used for deconvolution, 3) the range and size of the fluorescence bins (and grid space), 4) the internal logic of the objective function that is minimized (e.g., how the constraints are implemented, what value is minimized (RSS vs BIC), the primary minimization algorithm, the initial condition estimates, convergence and tolerance, etc.), and, 5) error and error propagation and noise considerations. While many of the above factors were considered for developing this algorithm, they were not collectively optimized as part of this research.

Improvements to the current deconvolution method can be made on multiple fronts. For example, while the stable distribution was selected amongst MATLAB’s build-in distributions (Appendix, Section 2-2), many of these distributions can be described by 2, 3, or 4 parameters but only the default form of the distribution in MATLAB was considered. More specifically, the 2-parameter log-logistic distribution model often had a very comparable BIC to the stable distribution for describing the microsensor fluorescence. The log-logistic distribution can also be described by 3 parameters, and extending it as such, could provide a better fit than the stable distribution. One must ensure that the interpolated fluorescence distributions accurately describe the true microsensor fluorescence across the full range of expected light dosages in PBR, or prediction errors can increase (Table 4-3, Figure 4.10, 4.11).
It may be necessary to experimentally determine a greater number of dosage curves and use a piece-wise interpolation scheme with different characteristic functions across the range of light dosages expected in the PBR. The piece-wise interpolation scheme would then allow the user to utilize different statistical distribution models that describe the fluorescence distribution across different light dosages and therefore eliminate the need to require one statistic distribution as performed in this research study.

Gating the appropriate microsensors events is another important factor that could influence the fluorescence distribution of the microsensors. Choosing the correct populations for fluorescence analysis (single microspheres and not doubles) is especially important if aggregates form since these aggregates will likely not fluoresce with the same pattern as a single particle.

Another important consideration when assessing how well, and with what certainty, this method can predict a light distribution is the random and systematic error and error propagation. Understanding the random and systematic error is important to consider alongside with dosage bin width to help ensure that the deconvolution predictions represent all possible dose distributions that could theoretically create a given fluorescence distribution and that the deconvolution algorithm is not 1) artificially converging to one and only one primary (potentially incorrect) solution and/or 2) overfitting. Thus, a major improvement to the current deconvolution process would involve implementing error components and binning constraints that take these background noise considerations into account when predicting solutions; including noise is commonly implemented in many photo-rending applications that use deconvolution to de-blur an image. On a similar note, establishing and quantifying the baseline uncertainty of the flow cytometer via technical replicates is critically important such
that any unnatural shifts in the recorded signal, which can be caused by fluctuations in ambient
temperature and/or light, vibrations from vortexing, and clogged lines and/air bubbles within
the sample lines, are quickly observed. Deconvolution is very sensitive to such changes, and,
if not caught, these blips could lead to egregious error in estimating the light distributions.
Great care and attentiveness must be taken while actively analyzing samples to observe and
correct any such disturbance.

**Proof-of-concept testing and validation**

*Estimating light intensity within a PBR*

In the current work, the estimates for the light intensity within the PBR were calculated
using minimal light intensity measurements (n=1 prediction made by n=3 measurements).
These calculations assumed that the light measured at one point along the PBR depth was
representative of the average light intensity across that volume section. The estimate of the
light intensity throughout the reactor volume could be improved with additional measurements
as well as by considering the transmittance and reflectance of the reactor material. These
improvements would provide a better validation tool for the microsensors.

*PBR experimental testing conditions*

A major limitation of the proof-of-concepts tests was in the use of DI water, without
microalgae. Since microalgae were not present, the gradient in light across the reactor depth
was minimized and only due to natural attenuation and scatter of light. In a real PBR, light
gradients are much sharper due to light absorbance from the microalgae’s photosynthesizing
pigments, and the scatter caused by their cells. Another important limitation of the current
work is that the microsensors were only tested with light wavelengths in the blue range (~400-
450 nm). The use of the microsensors in sunlight could prove more challenging as the dye is also activated by UV light and will likely uncage due to these high-energy wavelengths at a much faster rate; the impact of these other wavelengths on this dye were not quantified here.

Testing these microsensors in a more realistic setting than the current experiments is critical if they are to be used for scale-up purposes. An important additional test is to assess how mixing conditions within a PBR will impact the light exposure of microalgae. Gas bubbling (e.g., CO₂ or air) is frequently used in PBRs to aid in mixing and also to help maintain pH and dissolved CO₂ conditions necessary for microalgal growth. Gas bubbles increase light scattering within the fluid and can decrease the transmission throughout the reactor depth. This decrease in light intensity can be observed using a static sensor but this type of light measurements does not provide insight into how much of that scattered light is available for absorbance by adjacent particles in path of the scatter. Testing these microsensors within a PBR with different, realistic mixing conditions, e.g. changing impeller mixing speed, injecting gas into reactor at different flow rates and with different bubble sizes, could provide more realistic estimates that are more comparable to full-scale PBRs.

Summary and future applications

While further tests and improvement are needed to further validate this method, the overall results from this research show great promise. In summary, the current method could be further improved by, 1) increasing number of light intensity and dosage replicates to establish better model for distribution parameters, 2) determining the light dosage saturation point of the microsensors, 3) exploring other statistical distributions to describe microsensor fluorescence under collimated light outside those considered 4) re-structuring the
deconvolution algorithm to include noise and to optimize $BIC$ instead of $RSS$, and 5) testing the microsensors in varying reactor designs and configurations with more variable light distributions.

The ability to accurately quantify light within a reactor is of utmost importance when optimizing PBR designs, and current methods used to measure light make these predictions extremely challenging. The microsensors described herein offer an alternative to static sensors for quantifying the light distributions within a PBR and could aid significantly in optimizing and designing efficient microalgal cultivation systems for biofuel production.
REFERENCES


### Section 1: Supplemental tables and figures

Table A1-1: Supplemental coupling information for labeling dye with microspheres. Experimental reference for coupling dye to beads. Many calculations/values in table below are based on specifications as laid out in Table 3-1, Table 3-2, and Table 3-3.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of collimated experiments (#)</td>
<td>10</td>
</tr>
<tr>
<td>Volume required for single collimated experiments (ml)</td>
<td>20</td>
</tr>
<tr>
<td>Experimental microspheres concentration HIGH (#/ml)</td>
<td>1E+05</td>
</tr>
<tr>
<td>Experimental microspheres concentration LOW (#/ml)</td>
<td>5E+04</td>
</tr>
<tr>
<td>Total microspheres needed for experiments (#)</td>
<td>2E+07</td>
</tr>
<tr>
<td>Fraction of microsensor assumed recovered after single wash (fraction)</td>
<td>0.7</td>
</tr>
<tr>
<td>Total washing cycles (#)</td>
<td>3</td>
</tr>
<tr>
<td>Overall fraction of microspheres remaining after all washes (fraction)</td>
<td>0.34</td>
</tr>
<tr>
<td>Total number of microsphere needed for coupling (#)</td>
<td>5.83E+07</td>
</tr>
<tr>
<td>Total volume of microsphere stock solution (ml)</td>
<td>3.89E-02</td>
</tr>
<tr>
<td>Safety factor (NHS-ester excess: microsphere amine-surface group) (x)</td>
<td>5</td>
</tr>
<tr>
<td>Fraction of NHS-ester groups on microspheres for polymer (fraction)</td>
<td>0.5</td>
</tr>
<tr>
<td>Fraction of NHS-ester groups on microspheres for dye (fraction)</td>
<td>0.5</td>
</tr>
<tr>
<td>Total dye required (mg)</td>
<td>0.017</td>
</tr>
<tr>
<td>Total polymer required (mg)</td>
<td>0.037</td>
</tr>
<tr>
<td>DMSO: NHS Ratio (µL/mg)</td>
<td>50</td>
</tr>
<tr>
<td>DMSO for dye (µL)</td>
<td>0.84</td>
</tr>
<tr>
<td>DMSO for polymer (µL)</td>
<td>1.85</td>
</tr>
<tr>
<td>Microsphere volume to water ratio in coupling vial (ml/ml)</td>
<td>1.0E-02</td>
</tr>
<tr>
<td>Microsphere volume to buffer ratio in coupling vial (ml/ml)</td>
<td>6.7E-03</td>
</tr>
<tr>
<td>Total volume of microsphere soln. (ml)</td>
<td>3.9E-02</td>
</tr>
<tr>
<td>Total volume of microsphere added (ml)</td>
<td>7.4E-04</td>
</tr>
<tr>
<td>Total volume of water in microsphere stock soln. (ml)</td>
<td>3.8E-02</td>
</tr>
<tr>
<td>Total volume of DI water added (ml)</td>
<td>3.6E-02</td>
</tr>
<tr>
<td>Total buffer added (ml)</td>
<td>1.1E-01</td>
</tr>
<tr>
<td>Total water in coupling vial solution (ml)</td>
<td>7.4E-02</td>
</tr>
<tr>
<td>Total liquid volume in coupling vial (ml)</td>
<td>1.9E-01</td>
</tr>
<tr>
<td>Total microspheres mass (g)</td>
<td>7.9E-04</td>
</tr>
<tr>
<td>Total water mass in coupling vial (g)</td>
<td>1.9E-01</td>
</tr>
<tr>
<td>Total dye + DMSO mixture volume (µl)</td>
<td>8.4E-01</td>
</tr>
<tr>
<td>Total polymer + DMSO mixture volume (µl)</td>
<td>1.85</td>
</tr>
<tr>
<td>Total dye mass in coupling vial (mg)</td>
<td>1.7E-02</td>
</tr>
<tr>
<td>Total polymer mass in coupling vial (mg)</td>
<td>3.7E-02</td>
</tr>
<tr>
<td>Total dye (mMole)</td>
<td>3.0E-05</td>
</tr>
<tr>
<td>Total polymer (mMole)</td>
<td>3.0E-05</td>
</tr>
<tr>
<td>Total microspheres in coupling vial solution (#)</td>
<td>5.8E+07</td>
</tr>
<tr>
<td>Total microspheres in coupling vial solution (µm)</td>
<td>0.01</td>
</tr>
<tr>
<td>Total microsphere amine group molarity in coupling vial solution (µM)</td>
<td>65.4*</td>
</tr>
<tr>
<td>Total NHS-ester group molarity in coupling vial solution (mM)</td>
<td>0.33**</td>
</tr>
<tr>
<td>Parameter</td>
<td>Value</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>Solvent volume: microsphere volume</td>
<td>3.61</td>
</tr>
<tr>
<td>Total mass of coupling vial solution (mg)</td>
<td>0.19</td>
</tr>
<tr>
<td>% solids of microspheres in coupling vial (%)</td>
<td>0.42</td>
</tr>
<tr>
<td>Washing dilution factor (x)</td>
<td>3</td>
</tr>
</tbody>
</table>

**Wash #1**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer volume (ml)</td>
<td>0.56</td>
</tr>
<tr>
<td>Vortex time (min)</td>
<td>1 to 2</td>
</tr>
<tr>
<td>Centrifuge speed (rpm)</td>
<td>4000</td>
</tr>
<tr>
<td>Centrifuge time (min)</td>
<td>5 to 10</td>
</tr>
</tbody>
</table>

**Wash #2**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer volume (ml)</td>
<td>0.56</td>
</tr>
<tr>
<td>Vortex time (min)</td>
<td>1 to 2</td>
</tr>
<tr>
<td>Centrifuge speed (rpm)</td>
<td>4000</td>
</tr>
<tr>
<td>Centrifuge time (min)</td>
<td>5 to 10</td>
</tr>
</tbody>
</table>

**Wash #3**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DI water volume (ml)</td>
<td>0.56</td>
</tr>
<tr>
<td>Vortex time (min)</td>
<td>1 to 2</td>
</tr>
<tr>
<td>Centrifuge speed (rpm)</td>
<td>4000</td>
</tr>
<tr>
<td>Centrifuge time (min)</td>
<td>5 to 10</td>
</tr>
</tbody>
</table>

| Desired microsensor stock conc. (#/ml) | 5.0E+06 |
| Estimate microsphere number in tube after decant (#) | 2.0E+07 |

**suggested range 50-100 μM**: more dilute amine microsphere solutions result in excessive hydrolysis of the NHS-esters.

**suggested range: .1-10 mM**: exceeding this range could lead to dissolution of latex microspheres and/or plastic tubes.

- Filtered DI (0.22 μm) to add after decant (ml): 4.00
Figure A1-1: Spectral power distribution from blue LED lamp (LumiGrow) used in all experiments. Spectral light intensity of light used for testing microsensors. The total light intensity was calculated by integrating the spectral power distribution between 400 to 500 nm.

![Spectral Power Distribution](image1.png)

Figure A1-2: Absorbance and emission (caged and uncaged) spectrum for Abberior CAGE-552 NHS-ester dye

![Absorbance and Emission Spectrum](image2.png)

Section 2: MATLAB supplemental information

Section 2-2: Kernel density estimates for fluorescent distributions

\[ f = \text{ksdensity}(x, \text{pts}) \rightarrow \text{returns a probability density estimate, } f, \text{ for the data sample in the vector or two-column matrix } x, \text{ evaluated at the specified values in vector pts} \]

(https://www.mathworks.com/help/stats/ksdensity.html#btn1_p8-2)

EXAMPLE:

% Starting and stopping FI bin for deconvolution and FI bin width. The same FI grid space was used for all samples.
FI_Bin_Start = 0;
FI_Bin_Stop = 60000;
FI_Bin_Width = 200;

% Create equally spaced vector of fluorescence bins for kernel density estimate
FI_space = FI_Bin_Start;
FI_value_to_Add = FI_Bin_Start+FI_Bin_Width;
while FI_value_to_Add < FI_Bin_Stop
    FI_space = [FI_space FI_value_to_Add];
    FI_value_to_Add = FI_value_to_Add + FI_Bin_Width;
End

% Note, convolution_y is relative frequency of microsphere sample to fluorescence at said x fluorescence bin within FI_space
[convolution_y] = ksdensity(flow_FI_data_vector,FI_space);

***************************END EXAMPLE****************************
Section 2-1: Distribution testing with AllFitDist

Distribution fitting was done in MATLAB R2016b:

Function/Package Reference: AllFitDist* by Mike Sheppard 06 Feb 2012 (Updated 04 Apr 2012)

Description: ALLFITDIST Fit all valid parametric probability distributions to data.

Distributions tested:

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beta</td>
<td>Log-logistic</td>
</tr>
<tr>
<td>Birnbaum-Saunders</td>
<td>Lognormal</td>
</tr>
<tr>
<td>Exponential</td>
<td>Nakagami</td>
</tr>
<tr>
<td>Extreme value</td>
<td>Normal</td>
</tr>
<tr>
<td>Gamma</td>
<td>Rayleigh</td>
</tr>
<tr>
<td>Generalized extreme value</td>
<td>Rician</td>
</tr>
<tr>
<td>Generalized Pareto</td>
<td>Stable*</td>
</tr>
<tr>
<td>Inverse Gaussian</td>
<td>t location-scale</td>
</tr>
<tr>
<td>Logistic</td>
<td>Weibull</td>
</tr>
<tr>
<td>Negative binomial</td>
<td>Binomial</td>
</tr>
<tr>
<td>Poisson</td>
<td></td>
</tr>
</tbody>
</table>

*added manually to update default distributions to be consistent with those in MATLAB R2016B
Section 3: Python 3.2.1 Scripts for deconvolution

The following scripts/functions below were all created in Python 3.2.1 and written by Amanda L. Karam. *Denotes outside package/function used within script.

Script 1: Deconvolution without interpolation

Package/Function Reference: get_convolution_and_dose_matrix_and_output_file, main_function_no_interpolation, numpy*

Purpose: Runs multiple optimization problems for convolved samples using experimentally generated fluorescence distribution curves. Store optimization information to data file.

Requires: Manual inputs for variables that describe experimental data (date of experiment, fluorescence bin start, fluorescence bin end, fluorescence bin width, list of experimental convolution sample identifications, number of optimizations, method to use for minimize objective function). Requires that files describing experimental data are saved in correct format. Also requires imported functions as listed in script below.

Output: Data file with optimization solution and specifics regarding optimization problem. Also saves best fluorescence distribution curve generated for each sample ID input after specified number of optimizations

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

import get_convolution_and_dose_matrix_and_output_file
import main_function_no_interpolation
import numpy as np

# Input experiment information
experimentDate = '2016-07-19'
interpolationBooleanValue = False

# Fluorescence Channel to use? Which filter? Area or Width? fl2H means 585 filter, height
channelsForMinimizing = 'fl2H'

# Fluorescence Binning Information
fIBinWidth = 200
fIBinStart = 0
fIBinEnd = 60000
fIBinNum = (fIBinEnd - fIBinStart) / fIBinWidth

# Dose Binning Information for interpolation kJ-m2
doseList = [2, 4, 7, 9]
doseBinNum = len(doseList)
# Sample IDs for Convolution
convolutionSymbolList = ['C1', 'C2', 'C3']

# Specifics about how many optimizations attempts to use and what method to use for minimize objective function
iterationMax = 100
methodToMinimize = 'Powell'

# Create space to store all best FI distributions from all samples and compares these to actual
dose_fractions_best_all = np.zeros((doseBinNum, len(convolutionSymbolList)*2))
predicted_convoluted_fi_distribution_best_array_all = np.zeros((fIBinNum, len(convolutionSymbolList)*2+1))

global outputFileBest
global doseFractionBest

# Changing data format. Note, there is likely a way to create this data in format I want to start with....this seems inefficient but it works**
dose_increments_for_saving_data = np.arange(fIBinStart, fIBinEnd - 1, fIBinWidth)
i = 0
while i < len(dose_increments_for_saving_data):
    predicted_convoluted_fi_distribution_best_array_all[i, 0] = dose_increments_for_saving_data[i]
    i += 1

# Loop through all samples listed in convolution Symbol List and run deconvolution max iterations specified above
for k in range(0, len(convolutionSymbolList)):
    # start with first sample of day
    convolutionSymbol = convolutionSymbolList[k]
    # reset rss best number. For the # iterations of trials, this is going to be the all time best number to beat
    rssToBeat = 100000
    # Create a dictionary of experimental information easy to pass to functions
    experimentInformationNoInterpolation = {'Experiment Date': experimentDate,
                                             'Channel': channelsForMinimizing,
                                             'Convolution Sample ID': convolutionSymbol,
                                             'FI Bin Start': fIBinStart,
                                             'FI Bin End': fIBinEnd,
                                             'FI Bin Width': fIBinWidth,
                                             'FI Bin Number': fIBinNum,
                                             'Dose List': doseList}
Based on inputs and experiment information, there will be 2 csv file in directory that correspond (one of doses and one of convolution FI distribution). This function imports those files and converts it to the data format needed for deconvolution. This function also creates a text file with headers that all optimization attempts will be written to and returns the name of this file to pass along to next function.

doseAndConvolutionAndOutputFileList = 

generate_csv_from_files(experimentInformationNoInterpolation, interpolationBooleanValue)

doseMatrix = doseAndConvolutionAndOutputFileList[0]
convolutionMatrix = doseAndConvolutionAndOutputFileList[1]
outputFile = doseAndConvolutionAndOutputFileList[2]

Best predictions from each convolution will be saved to this file. Name modified later so it doesn't overwrite other data.

outputFileBest = outputFile

This step called the main function which sets up a new problem with random initial conditions, marks start time, calls the minimize function, specifying which method to use, saves data to output file along with date, sample ID, iteration number, etc.

iterationNumber = 0

while iterationNumber < iterationMax:
    outputFromMain = main_function_no_interpolation(iterationNumber, methodToMinimize, doseMatrix, convolutionMatrix, experimentInformationNoInterpolation, outputFile)

    rssCurrent = outputFromMain[0]
    doseFractionCurrent = outputFromMain[1]

    if rssCurrent < rssToBeat:
        doseFractionBest = doseFractionCurrent
        rssToBeat = rssCurrent

    print(convolutionSymbol, iterationNumber)

    iterationNumber += 1

# After # of iterations, the dose fractions that gave best overall match to actual data is converted into a matrix so it can be multiple with dose curves and generate best FI dist.
doseFractionBestMatrix = np.matrix(doseFractionBest)
doseFractionBestMatrix = doseFractionBestMatrix.transpose()

# Create the convolution output matrix based on best fractions. Stored this best data and actual data for particle
# convolution sample.
predicted_convoluted_fi_distribution_best = doseMatrix * doseFractionBestMatrix
predicted_convoluted_fi_distribution_best_array = np.asarray(predicted_convoluted_fi_distribution_best)
actual_convoluted_fi_distribution_array = np.asarray(convolutionMatrix)
jj = 0
while jj < len(predicted_convoluted_fi_distribution_best_array):
    predicted_convoluted_fi_distribution_best_array_all[jj, k + 1] = float(predicted_convoluted_fi_distribution_best_array[jj, 0])
    predicted_convoluted_fi_distribution_best_array_all[jj, k + 1 + len(establishmentList)] = float(actual_convoluted_fi_distribution_array[jj, 0])
    jj += 1

# After all samples specified in convolution list are done. The best output vs actual of each are saved as csv file.
output_name_best_fit = outputFileBest[0:len(outputFileBest)-4] + 'Best_vs_Actual.txt'
np.savetxt(output_name_best_fit, predicted_convoluted_fi_distribution_best_array_all, delimiter='","')

************************************************************************END SCRIPT************************************************************************
Script 2: Deconvolution with interpolation:

Package/Function Reference: get_convolution_and_dose_matrix_and_output_file, main_function_interpolation, numpy*

Purpose: Runs multiple optimization problems for convolved samples using experimentally generated fluorescence distribution curves.

Requires: Experimental data to be saved in parent directory in correct format. This data includes dose curves/bins interpolated based on experimental samples and kernel density estimates of convolved sample. Also, requires manual inputs for variables that describe the date of experiment, fluorescence bin start, fluorescence bin end, fluorescence bin width, list of experimental convolution sample identifications, number of optimizations, method to use for minimize objective function). Requires that files describing experimental data are saved in correct format (script below). Also requires imported functions as listed in script below.

Output: Data file with optimization solution and specifics regarding optimization problem. Also saves best fluorescence distribution curve generated for each sample ID input after specified number of optimizations

****************************START SCRIPT****************************

import get_convolution_and_dose_matrix_and_output_file
import main_function_interpolation
import numpy as np

# Input experimental information
# File date associated with experimental information. All data files from date
# should be stored in directory with this format.
experimentDate = '2016-07-29'

# Fluorescence Channel to use? Which filter? Area or Width? fl2H means 585 filter, height
channelsForMinimizing = 'fl2H'

# Fluorescence Binning Information
flBinWidth = 200
flBinStart = 0
flBinEnd = 60000
flBinNum = (flBinEnd - flBinStart) / flBinWidth

# Dose Binning Information for interpolation kJ-m2
doseBinWidth = 1
doseBinStart = 1
doseBinEnd = 16
doseBinNum = (doseBinEnd - doseBinStart) / doseBinWidth

71
# Sample IDs for Convolution
convolutionSymbolList = ['1min', '2min', '4min', '8min']

# Specifics about how many optimizations attempts to use and what method to use for minimize objective function
iterationMax = 100
methodToMinimize = 'Powell'

# Is interpolation used? If so, true, otherwise false.
interpolationBooleanValue = True

# Create space to store all best FI distributions from all samples and compares these to actual
dose_fractions_best_all = np.zeros((doseBinNum, len(convolutionSymbolList)*2))
predicted_convoluted_fi_distribution_best_array_all = np.zeros((fIBinNum, len(convolutionSymbolList)*2+1))

# Global variable for a file name where best possible solutions were found for each convolution sample ID
global outputFileBest

# Global variable declaration for an array to store best possible dose distribution values for each convolution sample ID
global doseFractionBest

# Changing data format.
dose_increments_for_saving_data = np.arange(fIBinStart, fIBinEnd - 1, fIBinWidth)
i = 0
while i < len(dose_increments_for_saving_data):
    predicted_convoluted_fi_distribution_best_array_all[i, 0] = dose_increments_for_saving_data[i]
    i += 1

# Loop through all samples listed in convolution Symbol List and run deconvolution max iterations specified above
for k in range(0, len(convolutionSymbolList)):

    # start with first sample of day
    convolutionSymbol = convolutionSymbolList[k]

    # reset rss best number. For the # iterations of trials, this is going to be the all time best number to beat
    rssToBeat = 100000

    # Create a dictionary of experimental information easy to pass to functions
    experimentInformationForInterpolation = {'Experiment Date': experimentDate,
doseAndConvolutionAndOutputFileList = \
get_convolution_and_dose_matrix_and_output_file(experimentInformationForInterpolation, interpolationBooleanValue)

# Based on inputs and experiment information, there will be 2 csv file in directory that correspond (one of doses
# and one of convolution FI distribution). This function imports those files and converts it to the data format
# needed for deconvolution. This function also creates a text file with headers that all optimization attempts will
# be written to and returns the name of this file to pass along to next function.
doseAndConvolutionAndOutputFileList = \
get_convolution_and_dose_matrix_and_output_file(experimentInformationForInterpolation, interpolationBooleanValue)

doseMatrix = doseAndConvolutionAndOutputFileList[0]
convolutionMatrix = doseAndConvolutionAndOutputFileList[1]
outputFile = doseAndConvolutionAndOutputFileList[2]

# Best predictions from each convolution will be saved to this file. Name modified later so it doesn't overwrite
# other data
outputFileBest = outputFile

# This step called the main function which sets up a new problem with random initial conditions, marks start time
# calls the minimize function, specifying which method to use, saves data to output file along with date, sample ID
# iteration number, etc.
iterationNumber = 0
while iterationNumber < iterationMax:
    outputFromMain = main_function_interpolation(iterationNumber, methodToMinimize, doseMatrix, convolutionMatrix,
experimentInformationForInterpolation, outputFile)

rssCurrent = outputFromMain[0]
doseFractionCurrent = outputFromMain[1]

if rssCurrent < rssToBeat:
    doseFractionBest = doseFractionCurrent
    rssToBeat = rssCurrent

print(convolutionSymbol, iterationNumber)

iterationNumber += 1
#

# After # of iterations, the dose fractions that gave best overall match to actual data is
# converted into a matrix so it can be multiple with dose curves and generate best FI dist.
doseFractionBestMatrix = np.matrix(doseFractionBest)
doseFractionBestMatrix = doseFractionBestMatrix.transpose()

# Create the convolution output matrix based on best fractions. Stored this best data and
# actual data for particle
# convolution sample.
predicted_convoluted_fi_distribution_best = doseMatrix * doseFractionBestMatrix
predicted_convoluted_fi_distribution_best_array =
np.asarray(predicted_convoluted_fi_distribution_best)

actual_convoluted_fi_distribution_array = np.asarray(convolutionMatrix)
jj = 0
while jj < len(predicted_convoluted_fi_distribution_best_array):
    predicted_convoluted_fi_distribution_best_array_all[jj, k + 1] = float(
        predicted_convoluted_fi_distribution_best_array[jj, 0])
    predicted_convoluted_fi_distribution_best_array_all[jj, k + 1 +
    len(convolutionSymbolList)] = float(
        actual_convoluted_fi_distribution_array[jj, 0])
    jj += 1

# After all samples specified in convolution list are done. The best output vs actual of each
# are saved as csv file.
output_name_best_fit = outputFileBest[0:len(outputFileBest)-4] + 'Best_vs_Actual.txt'
np.savetxt(output_name_best_fit, predicted_convoluted_fi_distribution_best_array_all,
delimiter=',',
********************************************************END SCRIPT********************************************************
**Function 1: get_convolution_and_dose_matrix_and_output_file**

**Package/Function Reference:** get_output_file_for_interpolation, get_output_file_for_no_interpolation, numpy*

**Purpose:** Set up data files for deconvolution. Imports convolution and dose distribution files based on experimental information dictionary and converts data into format needed for deconvolution. Also calls either Function 1 or Function 2 – based on whether or not interpolation specified – to create an output file that will store information about optimization inputs and outputs.

**Input Parameters:**

<table>
<thead>
<tr>
<th>experiment_information_dict</th>
<th>Dictionary of values which describe experimental and deconvolution information, will vary depending on whether or not interpolation is used.</th>
</tr>
</thead>
<tbody>
<tr>
<td>interpolation_boolean_value</td>
<td>Boolean value (True or False) that specifies whether problem uses interpolation or not. This will alter how output file is written, i.e. does it include output regarding BIC values, etc.</td>
</tr>
</tbody>
</table>

**Returns:** Returns a list containing dose distributions matrix, convolution vector, and string with the output file name.

***********START FUNCTION***********

```python
import numpy as np
import get_output_file_for_interpolation
import get_output_file_for_no_interpolation

def get_convolution_and_dose_matrix_and_output_file(experiment_information_dict, interpolation_boolean_value):
    convolution_sample_id = experiment_information_dict['Convolution Sample ID']
    channel = experiment_information_dict['Channel']
    experiment_date = experiment_information_dict['Experiment Date']

    fi_bin_start = experiment_information_dict['FI Bin Start']
    fi_bin_end = experiment_information_dict['FI Bin End']
    fi_bin_width = experiment_information_dict['FI Bin Width']

    fi_bin_info = '_FIBin_' + str(fi_bin_start) + '_to_' + str(fi_bin_end) + '_by_' + str(fi_bin_width)

    if interpolation_boolean_value:
        dose_bin_start = experiment_information_dict['Dose Bin Start']
        dose_bin_end = experiment_information_dict['Dose Bin End']
```

75
dose_bin_width = experiment_information_dict['Dose Bin Width']

dose_bin_info = '_DoseBin_' + str(dose_bin_start) + '_to_' + str(dose_bin_end) + '
    _by_' + str(dose_bin_width)
fi_bin_info = '_FIBin_' + str(fi_bin_start) + '_to_' + str(fi_bin_end) + '_by_' + str(fi_bin_width)

dose_curves_file = experiment_date + '\' + channel + fi_bin_info + dose_bin_info + '
    \_InterpolatedDoseCurves.csv'
convolution_file = experiment_date + '\' + channel + fi_bin_info + dose_bin_info + '
    \_Convolution_' + convolution_sample_id + '.csv'
output_file = get_output_file_for_interpolation(experiment_information_dict)
else:
dose_curves_file = experiment_date + '\' + channel + fi_bin_info + '
    \_DoseCurves.csv'
output_file = get_output_file_for_no_interpolation(experiment_information_dict)
convolution_file = experiment_date + '\' + channel + fi_bin_info + '
    \_Convolution_' + convolution_sample_id + '.csv'

# Import data sets as data frame
dose_curves = pd.read_csv(dose_curves_file, sep='\', header=None)
convolution = pd.read_csv(convolution_file, sep='\', header=None)

# Change dataframes to numpy arrays
dose_curves_array = dose_curves.as_matrix(columns=None)
convolution_array = convolution.as_matrix(columns=None)

# Change numpy arrays to matrices, note to self, surely there is a better way to import
data directly as matrices.
dose_curve_matrix = np.matrix(dose_curves_array)
convolution_matrix = np.matrix(convolution_array)
return [dose_curve_matrix, convolution_matrix, output_file]

*****************************************************************************END FUNCTION*****************************************************************************
Function 2: get_output_file_for_no_interpolation

Purpose: Generate output file and make sure header is written to file for deconvolution WITHOUT interpolation.

Input Parameters:

| experiment_information_dict: | Dictionary of values which describe experimental and deconvolution information. |

Returns: Return the name of output file.

*******************************************************************START FUNCTION*******************************************************************

def get_output_file_for_no_interpolation(experiment_information_dict):
    convolution_sample_id = experiment_information_dict['Convolution Sample ID']
    channel = experiment_information_dict['Channel']
    experiment_date = experiment_information_dict['Experiment Date']
    dose_list = experiment_information_dict['Dose List']
    fi_bin_start = experiment_information_dict['FI Bin Start']
    fi_bin_end = experiment_information_dict['FI Bin End']
    fi_bin_width = experiment_information_dict['FI Bin Width']

    fi_bin_info = '_FIBin_' + str(fi_bin_start) + '_to_' + str(fi_bin_end) + '_by_' + str(fi_bin_width)

    # File that all minimization results will be written (created based on inputs)
    output_file_name = experiment_date + '_Results\' + channel + '_DeconvolutionSolutions.txt'
    header_string = "Date, Convolution ID, Minimizing Channel, Method, FI Bin Start, FI Bin Stop, FI Bin Width, RSS, RMS, Sum of Doses, Computation time (s), " \n    "IterationNumber, Dose Bin, Dose Value, Average Dose" + "\n"

    # checks to see if output file has already been created with similar input conditions, if so data will be written # to that file. If not, a new file will be created.
    output_file = open(output_file_name, "a+")
    output_file.close()

    output_file = open(output_file_name, "r")
    first_line = output_file.readline()
    first_word = first_line[0:4]
    output_file.close()

    if first_word != 'Date':
        output_file = open(output_file_name, "a+")
output_file.write(header_string)
output_file.close()

return output_file_name

END FUNCTION

END FUNCTION
Function 3: get_output_file_for_interpolation

*Purpose:* Generate output file and make sure header is written to file for deconvolution WITH interpolation.

**Input Parameters:**

| experiment_information_dict: | Dictionary of values which describe experimental and deconvolution information. |

**Returns:** Return the name of output file.

********************************************************************************************

**START FUNCTION******************************************************************************************

```python
def get_output_file_for_interpolation(experiment_information_dict):
    convolution_sample_id = experiment_information_dict['Convolution Sample ID']
    channel = experiment_information_dict['Channel']
    experiment_date = experiment_information_dict['Experiment Date']
    fi_bin_start = experiment_information_dict['FI Bin Start']
    fi_bin_end = experiment_information_dict['FI Bin End']
    fi_bin_width = experiment_information_dict['FI Bin Width']
    dose_bin_start = experiment_information_dict['Dose Bin Start']
    dose_bin_end = experiment_information_dict['Dose Bin End']
    dose_bin_width = experiment_information_dict['Dose Bin Width']
    fi_bin_info = '_FIBin_ ' + str(fi_bin_start) + '_to_' + str(fi_bin_end) + '_by_' + str(fi_bin_width)
    dose_bin_info = '_DoseBin_ ' + str(dose_bin_start) + '_to_' + str(dose_bin_end) + '_by_' + str(dose_bin_width)
    output_file_name = experiment_date + '
\Results
' + channel + '
_FDeconvolutionSolutions.txt"
    header_string = "Date, Convolution ID, Minimizing Channel, Method, "
                     "Dose Bin Start, Dose Bin End, Dose Bin Width, "
                     "FI Bin Start, FI Bin Stop, FI Bin Width, RSS, RMS, Sum of Doses, Computation time (s), "
                     "Iteration Number, Dose Bin, Dose Value, AIC, BIC, Average Dose" + "\n"
    # checks to see if output file has already been created with similar input conditions, if so data will be written
    # to that file. If not, a new file will be created.
    output_file = open(output_file_name, "a+")
```
output_file.close()

# open file and read the first 5 characters of the first line.
output_file = open(output_file_name, "r")
first_line = output_file.readline()
first_word = first_line[0:4]
output_file.close()

# Determine if this first word says date, if it doesn’t this mean the header hasn’t been written to file yet. so
# write the header.
if first_word != 'Date':
    output_file = open(output_file_name, "a+")
    output_file.write(header_string)
    output_file.close()
return output_file_name

************************************************************************END FUNCTION************************************************************************
**Function 4: main_function_no_interpolation**

*Package/Function Reference: objective_function, numpy*, time*, lmfit* (minimize, Parameters)*

**Purpose:** Sets up new optimization problem with new initial dose parameter estimates for deconvolution WITHOUT interpolation. Calls minimize function to minimize objective function. Saves conditions surrounding optimization and optimization output to a file.

**Input Parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>iteration_number</td>
<td>iteration number for optimization trial</td>
</tr>
<tr>
<td>method_to_use</td>
<td>method to use for minimizing objective function,</td>
</tr>
<tr>
<td>dose_matrix</td>
<td>m x n matrix describing n fluorescence distributions vectors with m fluorescence bins. The fluorescence distributions vectors are generated experimentally.</td>
</tr>
<tr>
<td>convolution_matrix</td>
<td>m x 1 vector describing kernel density estimate of fluorescence distribution from sample</td>
</tr>
</tbody>
</table>

**Returns:** Residual squared sum optimization problem, i.e. the difference between the kernel density estimate for fluorescence distribution and estimate predicted by deconvolution.

****************************START FUNCTION***************************

```python
import numpy as np
import time
from lmfit import minimize, Parameters
import objective_function

def main_function_no_interpolation(iteration_number, method_to_use, dose_matrix, convolution_matrix, experiment_information_dict, output_file_name):
    convolution_sample_id = experiment_information_dict['Convolution Sample ID']
    channel = experiment_information_dict['Channel']
    experiment_date = experiment_information_dict['Experiment Date']
    fi_bin_start = experiment_information_dict['FI Bin Start']
    fi_bin_end = experiment_information_dict['FI Bin End']
    fi_bin_width = experiment_information_dict['FI Bin Width']
    dose_list = experiment_information_dict['Dose List']
    dose_matrix_shape = dose_matrix.shape
    number_of_dose_bins = float(dose_matrix_shape[1])
    number_of_fi_bins = float(dose_matrix_shape[0])
```
# Create random initial estimates for doses, these estimates will be added to parameter variable
initial_dose_estimate_array = np.random.random(number_of_dose_bins)

# Make sure initial estimate sum to 1
initial_dose_estimate_array /= initial_dose_estimate_array.sum()

# Make list of parameter names, alpha0, alpha1, alpha2, etc. this will depend on number of dose bins/courses
list_of_parameter_names = []
i = 0
while i < number_of_dose_bins:
    parameter_name = "alpha" + str(i)
    list_of_parameter_names.append(parameter_name)
    i += 1

# Convert dose information into matrix and then transpose matrix into correct shape
initial_dose_estimate_matrix = np.matrix(initial_dose_estimate_array)
initial_dose_estimate_matrix = initial_dose_estimate_matrix.transpose()

# Add all parameters to solve for and an initial estimate for each one. These values can vary and are constrained
# to value between 0-1; this is arbitrary constraint because in objective function all parameters are divided by
# the sum of total number of parameters
parameter_object_to_optimize = Parameters()
i = 0
while i < number_of_dose_bins:
    parameter_object_to_optimize.add(list_of_parameter_names[i],
    value=initial_dose_estimate_matrix[i, 0],
    min=0, max=1)
    i += 1

# mark start time of optimization
start_time = time.time()

# Call minimize function
# Minimize function that will minimize objective function.
output_for_optimization = minimize(objective_function, parameter_object_to_optimize,
    args=(dose_matrix, convolution_matrix, list_of_parameter_names),
    method=method_to_use)

# get the parameters from the optimization output
optimized_parameters = output_for_optimization.params

# get the sum of residuals of square error
rss = float(sum(output_for_optimization.residual))

# room mean squared
rms = rss/number_of_fi_bins

# create array to save parameter data to
dose_prediction_array = np.zeros((len(optimized_parameters), 1))
i = 0
while i < len(optimized_parameters):
    dose_value_raw = optimized_parameters[list_of_parameter_names[i]].value
    dose_prediction_array[i] = dose_value_raw
    i += 1

# remember, implement constraint fractions
dose_prediction_array /= dose_prediction_array.sum()

# Not sure why I change the format again, but don't want to mess with code, hmmm....?
dose_parameter_estimate = [0] * (len(optimized_parameters))
sum_of_parameters = 0
j = 0
while j < len(dose_parameter_estimate):
    dose_value = float(dose_prediction_array[j])
    dose_parameter_estimate[j] = dose_value
    sum_of_parameters += dose_value
    j += 1

# Calculate values that you want to save to files, computation time, dose values, etc.

# How long optimization took
computational_time = (time.time() - start_time)

# Calculate the average dose
average_dose = 0.0
i = 0
while i < (len(dose_parameter_estimate)):
    dose_fraction_value = (float(dose_parameter_estimate[i]))
    dose_bin_number = dose_list[i]
    dose_component_to_average = dose_bin_number * dose_fraction_value
    average_dose += dose_component_to_average
    i += 1

# Save data in long data format for easier graph making
i = 0
output_file = open(output_file_name, "a+")
while i < (len(dose_parameter_estimate)):
    dose_fraction_value = str(float(dose_parameter_estimate[i]))
dose_bin_number = str(dose_list[i])

string_of_output_to_write_to_file = experiment_date + ', ' + 
    convolution_sample_id + ', ' + 
    channel + ', ' + 
    method_to_use + ', ' + 
    str(fi_bin_start) + ', ' + 
    str(fi_bin_end) + ', ' + 
    str(fi_bin_width) + ', ' + 
    str(rss) + ', ' + 
    str(rms) + ', ' + 
    str(float(sum_of_parameters)) + ', ' + 
    str(computational_time) + ', ' + 
    str(iteration_number) + ', ' + 
    str(dose_bin_number) + ', ' + 
    str(dose_fraction_value) + ', ' + 
    str(average_dose)

    output_file.write(string_of_output_to_write_to_file + '\n')

    i += 1

output_file.close()

return [rss, dose_parameter_estimate]

**************************************************************************END FUNCTION**************************************************************************
Function 5: main_function_interpolation

Package/Function Reference: objective_function, numpy*, time*, lmfit* (minimize, Parameters), math*

Purpose: Sets up new optimization problem with new initial dose parameter estimates for deconvolution WITH interpolation. Calls minimize function to minimize objective function. Saves conditions surrounding optimization and optimization output to a file.

Input Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>iteration_number</td>
<td>iteration number for optimization trial</td>
</tr>
<tr>
<td>method_to_use</td>
<td>method to use for minimizing objective function</td>
</tr>
<tr>
<td>dose_matrix</td>
<td>m x n matrix describing n fluorescence distributions vectors with m fluorescence bins. The fluorescence distributions vectors are generated by interpolation function in MATLAB.</td>
</tr>
<tr>
<td>convolution_matrix</td>
<td>m x 1 vector describing kernel density estimate of fluorescence distribution from sample</td>
</tr>
</tbody>
</table>

Returns: Residual sum of squares for the optimization problem, i.e. the difference between the kernel density estimate for fluorescence distribution and estimate predicted

*****************************START FUNCTION***************************

import numpy as np
import time as time
from lmfit import minimize, Parameters
import objective_function
import math as math

def main_function_interpolation(iteration_number, method_to_use, dose_matrix, convolution_matrix, experiment_information_dict, output_file_name):
    convolution_sample_id = experiment_information_dict['Convolution Sample ID']
    channel = experiment_information_dict['Channel']
    experiment_date = experiment_information_dict['Experiment Date']
    fi_bin_start = experiment_information_dict['FI Bin Start']
    fi_bin_end = experiment_information_dict['FI Bin End']
    fi_bin_width = experiment_information_dict['FI Bin Width']
    dose_bin_start = experiment_information_dict['Dose Bin Start']
    dose_bin_end = experiment_information_dict['Dose Bin End']
    dose_bin_width = experiment_information_dict['Dose Bin Width']
    dose_matrix_shape = dose_matrix.shape
    number_of_dose_bins = float(dose_matrix_shape[1])
number_of_fi_bins = float(dose_matrix_shape[0])

# create random initial estimates for doses, these estimates will be added to parameter variable
initial_dose_estimate_array = np.random.random(number_of_dose_bins)
# make sure initial estimate sum to 1
initial_dose_estimate_array /= initial_dose_estimate_array.sum()

# Make list of parameter names, alpha0, alpha1, alpha2, etc. this will depend on number of doses
list_of_parameter_names = []
i = 0
while i < number_of_dose_bins:
    parameter_name = "alpha" + str(i)
    list_of_parameter_names.append(parameter_name)
    i += 1

initial_dose_estimate_matrix = np.matrix(initial_dose_estimate_array)
initial_dose_estimate_matrix = initial_dose_estimate_matrix.transpose()

# Add all parameters to solve for and an initial estimate for each one. These values can vary and are constrained
# to value between 0-1
parameter_object_to_optimize = Parameters()
i = 0
while i < number_of_dose_bins:
    parameter_object_to_optimize.add(list_of_parameter_names[i],
                                       value=initial_dose_estimate_matrix[i, 0],
                                       min=0, max=1)
    i += 1

# mark start time of optimization
start_time = time.time()

# Call minimize function
# Minimize function that will minimize objective function.
output_for_optimization = minimize(objective_function, parameter_object_to_optimize,
                                    args=(dose_matrix, convolution_matrix, list_of_parameter_names),
                                    method=method_to_use)
optimized_parameters = output_for_optimization.params

# create array to save parameter data to
dose_prediction_array = np.zeros((len(optimized_parameters), 1))
i = 0
while i < len(optimized_parameters):
    dose_value_raw = optimized_parameters[list_of_parameter_names[i]].value
dose_prediction_array[i] = dose_value_raw
i += 1

# remember, implement constraint fractions
dose_prediction_array /= dose_prediction_array.sum()

# Not sure why I change the format again, but don't want to mess with code, just inefficient code here
dose_parameter_estimate = [0] * (len(optimized_parameters))

# Just a check put in place to make sure the doses sum to one.
sum_of_parameters = 0
i = 0
while i < len(dose_parameter_estimate):
    dose_value = float(dose_prediction_array[i])
    dose_parameter_estimate[i] = dose_value
    sum_of_parameters += dose_value
    i += 1

# Calculate values that you want to save to files#

# how long optimization took
computational_time = (time.time() - start_time)

# Calculate the average dose
average_dose = 0.0
i = 0
while i < (len(dose_parameter_estimate)):
    dose_fraction_value = (float(dose_parameter_estimate[i]))
    dose_bin_number = (dose_bin_start + dose_bin_width * i)
    dose_component_to_average = dose_bin_number * dose_fraction_value
    average_dose += dose_component_to_average
    i += 1

# residual square sum
rss = float(sum(output_for_optimization.residual))

# root mean squared
rms = rss/number_of_fi_bins

# AIC criteria

# BIC criteria
bic = number_of_fi_bins + number_of_fi_bins*math.log10(2*math.pi) + \
number_of_fi_bins*math.log10(rss/number_of_fi_bins) +
math.log10(number_of_fi_bins)*(number_of_dose_bins + 1)

output_file = open(output_file_name, "a+")

# Save data in long format for easier graph making
i = 0
while i < (len(dose_parameter_estimate)):
    dose_fraction_value = str(float(dose_parameter_estimate[i]))
    dose_bin_number = str(dose_bin_start + dose_bin_width * i)

    string_of_output_to_write_to_file = experiment_date + ', ' + 
    convolution_sample_id + ', ' + 
    channel + ', ' + 
    method_to_use + ', ' + 
    str(dose_bin_start) + ', ' + 
    str(dose_bin_end) + ', ' + 
    str(dose_bin_width) + ', ' + 
    str(fi_bin_start) + ', ' + 
    str(fi_bin_end) + ', ' + 
    str(fi_bin_width) + ', ' + 
    str(rss) + ', ' + 
    str(rms) + ', ' + 
    str(sum_of_parameters) + ', ' + 
    str(computational_time) + '', '' + 
    str(iteration_number) + '', '' + 
    str(dose_bin_number) + '', '' + 
    str(dose_fraction_value) + '', '' + 
    str(aic) + '', '' + 
    str(bic) + '', '' + 
    str(average_dose)

    output_file.write(string_of_output_to_write_to_file + '\n')

    i += 1

output_file.close()

return [rss, dose_parameter_estimate]

******************************************************************************END FUNCTION******************************************************************************
**Function 6: objective_function**

**Purpose:** Function to be minimized. Calculates square error between actual fluorescence distribution from flow cytometer and fluorescence distribution predicted based on predicted dose distribution

**Input Parameters:**

<table>
<thead>
<tr>
<th>param</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>params</td>
<td>Parameter object containing parameters to be estimated</td>
</tr>
<tr>
<td>parameter_names</td>
<td>List of strings describing parameter names</td>
</tr>
<tr>
<td>dose_fi_distribution_matrix</td>
<td>m x n matrix describing n fluorescence distributions vectors with m fluorescence bins. The fluorescence distributions vectors are generated by interpolation function (Appendix C4) from interpolated curves</td>
</tr>
<tr>
<td>convoluted_fi_distribution</td>
<td>m x 1 vector describing kernel density estimate of fluorescence distribution from sample</td>
</tr>
</tbody>
</table>

**Returns:** Residual squared sum optimization problem, i.e. the difference between the kernel density estimate for fluorescence distribution and estimate predicted

```
def objective_function(params, dose_fi_distribution_matrix, convoluted_fi_distribution, parameter_names):
    dose_fractions = np.zeros((len(params), 1))
    j = 0
    while j < len(dose_fractions):
        dose_fractions[j, 0] = params[parameter_names[j]].value
        j += 1

    # Force dosage fractions equal 1! This is how the constraint is implemented.
    # Note, probably a better, more efficient way to implement constraint...(?)
    dose_fractions /= dose_fractions.sum()

    # Convert into a matrix so you can multiply with dosage matrix
    dose_fractions_matrix = np.matrix(dose_fractions)

    # Predicted convoluted signal based on estimated dose fraction and dosage curves
    predicted_convoluted_fi_distribution = dose_fi_distribution_matrix * dose_fractions_matrix

    # Different between convoluted file and predicted
    error = (predicted_convoluted_fi_distribution - convoluted_fi_distribution)
```
size_of_error_vector = len(error[:, 0])

# Create square error vector
square_error = np.zeros((size_of_error_vector, 1))

# Square Difference
i = 0
while i < len(error[:, 0]):
    square_error[i] = (error[i]) ** 2
    i += 1

return square_error

**************************************************************************END FUNCTION**************************************************************************