Nitrogenous Fate

Susan Garcia, Aaron Lecciones, Brian Roman, Carlyn Schmidgall

CSE583 Software Development for Data Scientists
University of Washington, Seattle

Autumn 2023
13 December 2023

GitHub

CSE 583 UW Fall AY 2023-24

Presentation Outline

1. Problem Statement
2. Proposed Solution
3. Software Design
    - Target Users
    - Functional Design
    - Technology Review
4. Project Design
    - Github Repository
    - Project Directory Structure
    - Components
    - Scripts
    - Testing
5. Results

Problem Statement

Biological oceanographers and marine biologists who study environmental samples in their microbial system experiments to characterize cellular phenotype and biochemical pathways are ***faced with the effects of obscuring variation*** when normalization is not conducted during measuring the rate of intracellular and dissolved metabolite production.

One normalization method employed is the best-matched internal standard (B-MIS) normalization which **is a step-heavy process** (Boysen *et al* 2018). Data visualization is also another step conducted as part of the process (Sacks *et al* 2022).

Proposed Solution

The team shall write code to handle data normalization and data visualization:

- Python Script for normalization of data and calculation of peak areas.
  - Panda
- Jupyter Notebook for data visualization
  - Seaborn for exploratory data analysis
  - Altair to develop interactive plots

Target Users

User Characterization for NF Software

The Nitrogenous Fate Software uses metabolite data of isotopically-labeled molecules (Nitrite, Ammonia, and Urea) through incubations from samples in the Equatorial Pacific to show pathways of nitrogen within communities through simple data visualization.

=======

User	Needs and Desires	Skill Level	Use Cases
Research Scientist (Marine Biologist)	Track isotopically-labeled Nitrogen within marine communities. Desires a simple yet parameter-configurable interface with multiple options for visualization.	Highly skilled scientist who is familiar with basic data science tools	Run program as is
Research Scientist (Generic)	Use software to track defined elements in a known system. Desires a simple yet highly parameter-configurable interface with multiple options for visualization.	Highly skilled scientist who is familiar with basic data science tools and Github coding	Run program with customization
Research Scientist (administrator)	Maintain, improve, and update program as needed. Desires a simple system reporting tool for user feedback.	Highly skilled scientist who is familiar with the coding used for the program	Run and add code or debug program
Researcher	Use software but without much knowledge of underlying mechanics. Desires a way to use the software without much research or professional background in the domain	Adequately skilled researcher who might not be familiar with data science tools	Run program with assistance

Document Information: CSE583 NF Project Documentation for User Characterization

Functional Design

Version 0.1 of this software implements only the following functional design for nitrogenousfate.py:

Functional Design	Use Case	Description	Prompt
Run Program	Input Data Set	Loads data file based on run command checks for data format	May raise exceptions when encountered
	Data Set Processing	scripts runs using either customization or without customizations.	
		Section I: Data Cleaning and Organization	
		Section II: Best Matched Internal Standard Normalization	
		Data analysis encounters an error	raise Exception or display Error, exit program
		Data analysis is successful	Display: 'Analysis completed' save output files.
End Program	Terminate Program	Safely terminate program and show credits	Display thank you prompt with credits, then exit app.

Section III is handled by a Jupyter Notebook.

Document Information: CSE583 NF Project Documentation for Functional Design

Technology Review

PANDAS

pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.

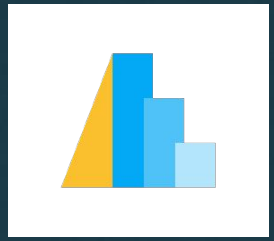
SEABORN

Positives:

- Standard plots out of the box
- Perfect for statistical analysis
- Fast to use for standard plots

Negatives:

- Built on top of matplotlib
- Less ability to customize



ALTAIR

Built on top of Vega and Vega-Lite grammars

Positives:

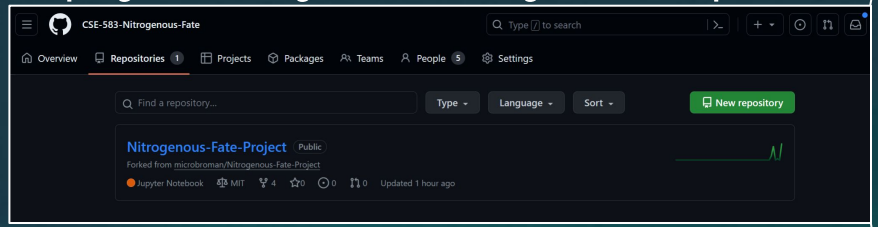
- Intuitive and structured approach to plotting
- Altair is interactive (zoom in, pan and grab, tooltips, etc)
- Flexible

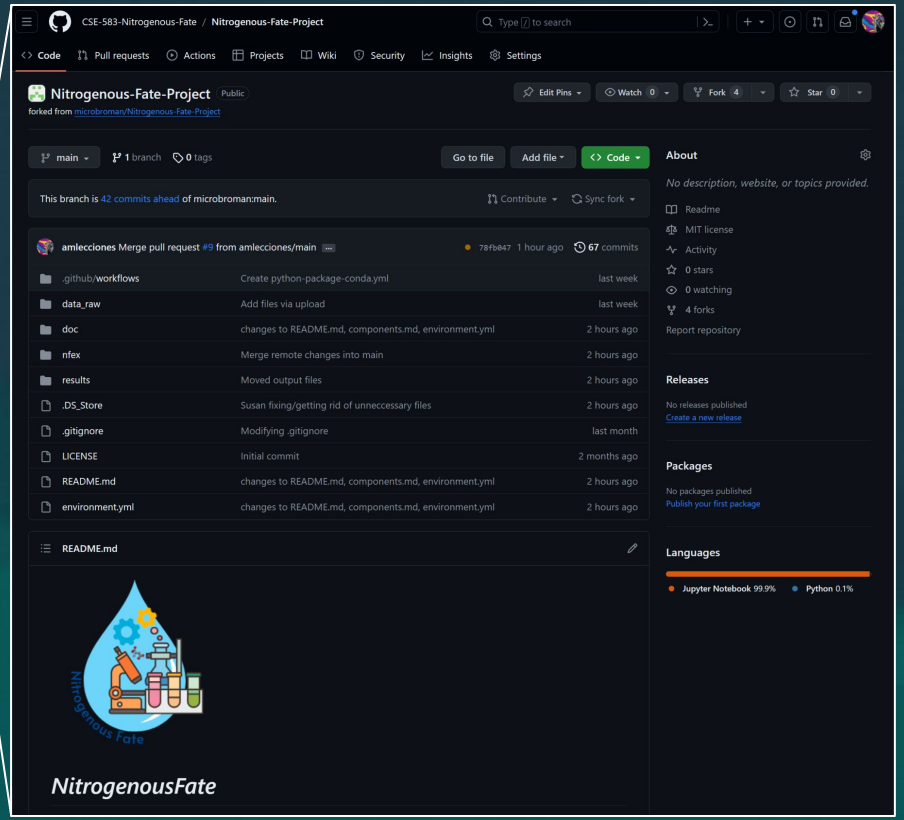
Negatives:

- No 3D plotting
- Not as customizable

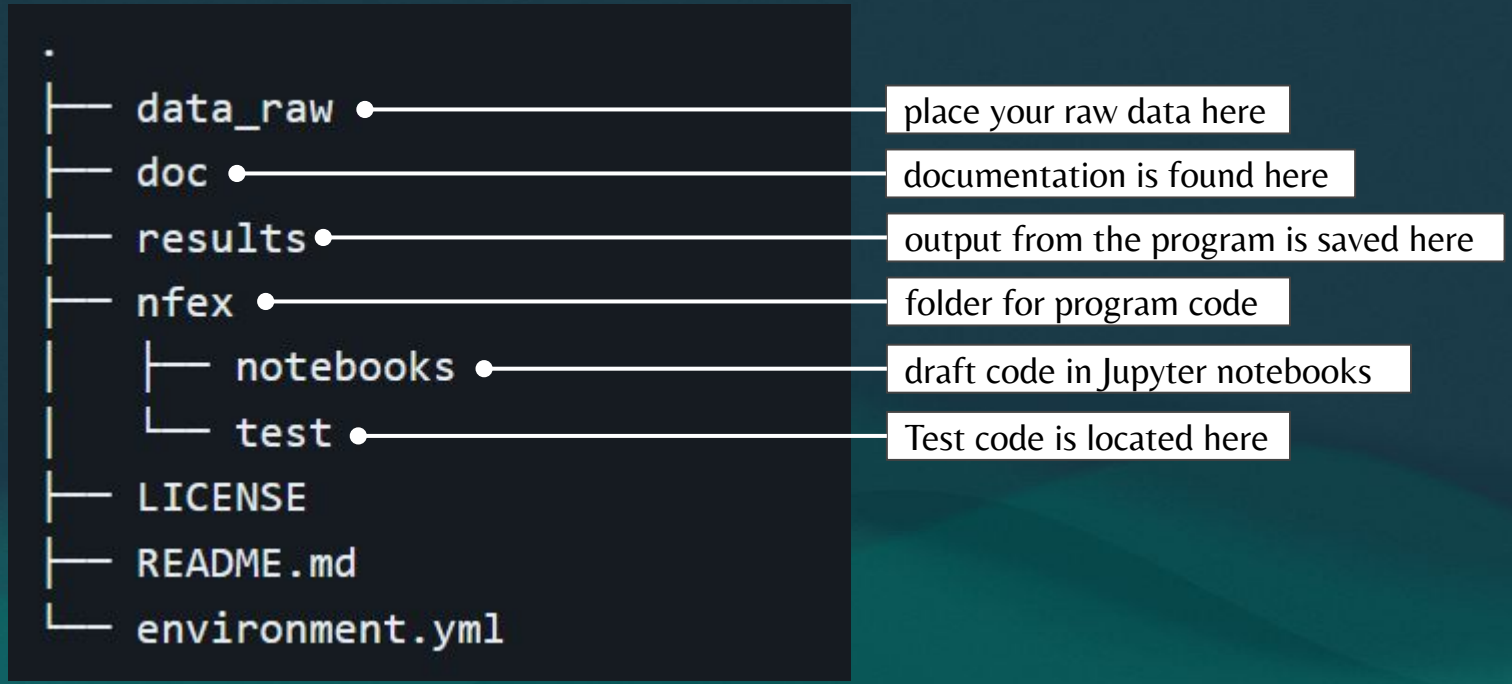
Project Design - Github Repository

https://github.com/orgs/CSE-583-Nitrogenous-Fate/repositories



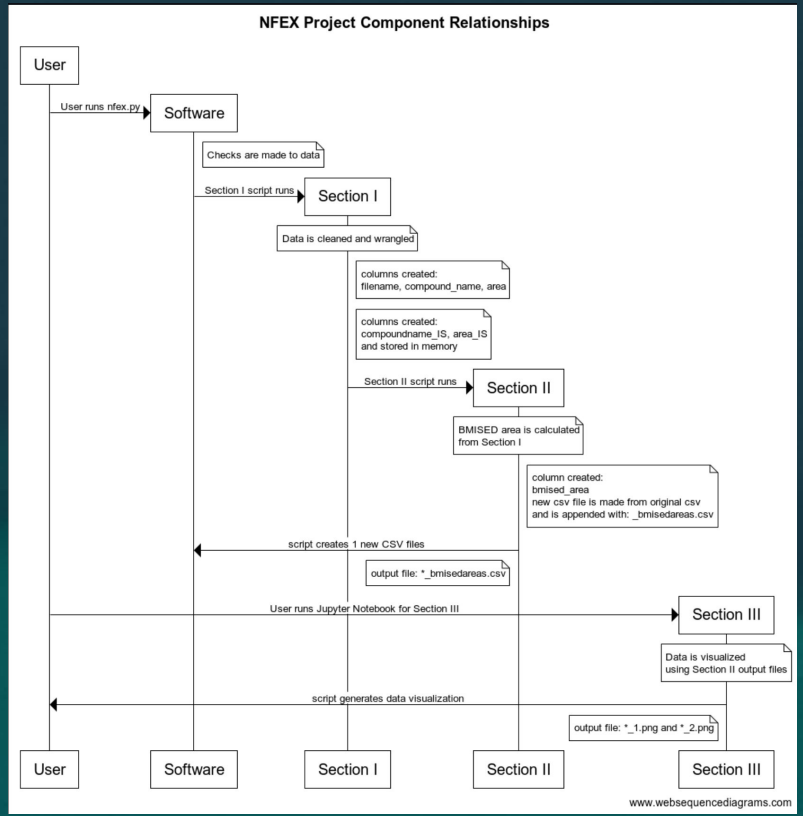


Project Design - Project Directory Structure



Project Design - Components

Name	Function	Inputs	Output	Interactions
Data Cleaning and Wrangling (Section I)	A `dataset` is loaded (either dissolved, exo or particulate, endo) and this component looks for relevant data (by column) from the data set. The script then calculates values for the following new columns which are appended to the `dataset` under a new file named `*_clean_areas.csv` : `filename`, `compound_name`, and `area`. It then computes an internal standard which is used to normalize data for use in section II, it creates two new columns for these values: `compoundname_IS` and `area_IS`, which are appended to the `dataset` under a new file names `*_IS_list.csv`.	run nfex (nitrogenousfate.py) `Section I script` with data in `data_raw` folder.	component will output values for: `filename`, `compound_name`, `area`, `compoundname_IS` and `area_IS`.	User monitors for any errors.
BMISED Computation (Section II)	The new data in memory which now contains values in the new columns: `filename`, `compound_name`, `area`, `compoundname_IS` and `area_IS` is used to calculate BMISED values. These values are placed in a new column: `bmised_area` and is saved in a dataset named `*_bmisedareas.csv`.	nfex `Section II script` runs using new data stored in memory.	component will output the `new dataset` : `*_bmisedareas.csv` for values in the new column: `bmised_area` in `results` folder.	User monitors for any errors expects a CSV file in `results` folder.
Data Visualization (Section III)	The `new dataset` which now contains values in the new column `bmised_area` is used to generate data visualization. The Jupyter Notebook generates multiple data visualizations to choose from. The user may save them into files.	run Jupyter Notebook (XXXX.ipynb) `Section III script` with correct data in `results` folder.	component will output image files from user input in Jupyer Notebook script.	User monitors for any errors, uses Jupyter Notebook to generate and save visualization files.



Project Design - Scripts

Version 0.1

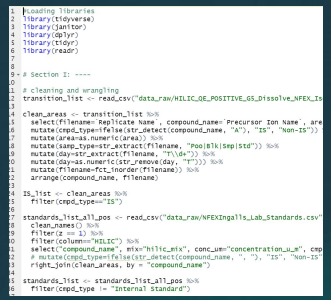
R Script

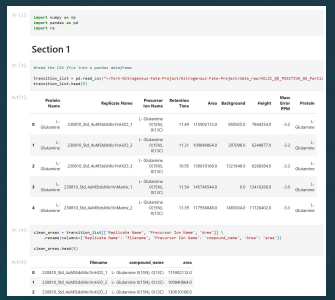
Jupyter Notebook

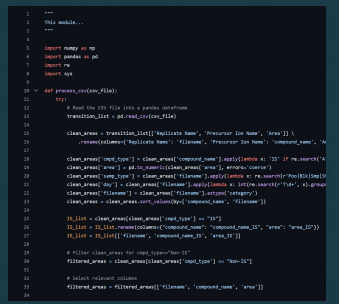
Python Script

Section I & II

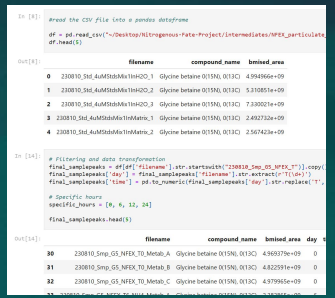
Computation for BMIS







Data Visualization



Future Versions

Project Design - Testing

Unit Tests performed to ensure correct data formatting, types in CSV file passed into data processing scripts.

Tests ensure:

- Viable, non-empty CSV is supplied
- Data types within CSV is correct (numeric)
- CSV contains internal standards to allow for quality control of data


```
"""
This module contains the unit test functions for the nitrogenous fate (nfex)
project

Classes:
- TestKFEX(unittest.TestCase)
"""

import unittest
from nitrogenousfate import process_csv

class TestNFEX(unittest.TestCase):
    """
    TestNFEX(unittest.TestCase): tests the code for data cleaning and wranging for NFEX.
    """

    def test_smoke_pass(self):
        """
        test_smoke_pass(self): Smoke test (1) verifying function should run without
        crashing or throwing errors if given an appropriate CSV file.
```

Results

Altair visualization:

- All data points in one graph
- Interactive graph - can select any compound from the legend
- Tooltip - can see information for any datapoint

Nitrogenous Fate

Susan Garcia, Aaron Lecciones, Brian Roman, Carlyn Schmidgall

CSE583 Software Development for Data Scientists
University of Washington, Seattle

Autumn 2023

Thank you For listening!

GitHub