Nutrient Profile Clustering

Laurel Meke
Nate Lauffenburger
Xinjin Li
Mariam Garcia
Sara Delawalla

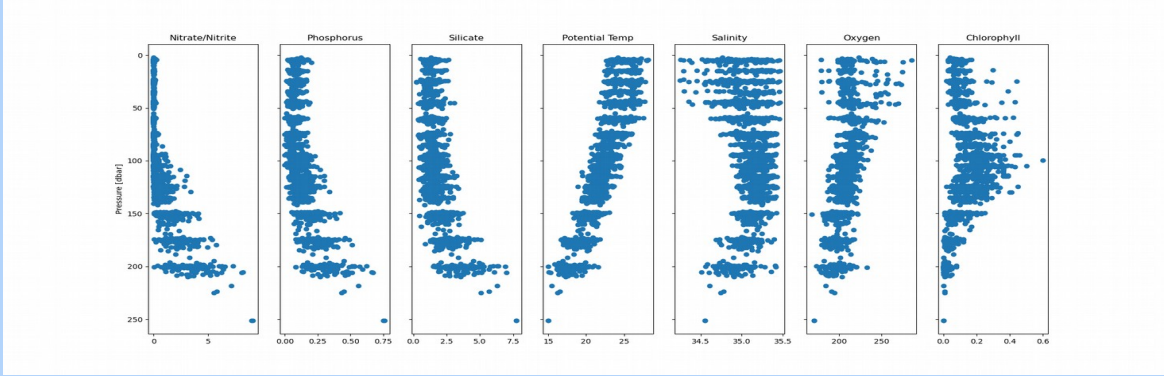
The Problem

We aimed to:

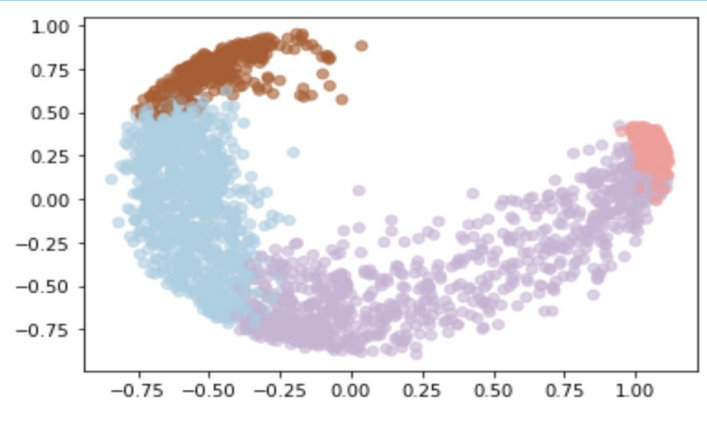
- Identify clusters in the distribution of nutrients across a certain location
- Identify clusters in the distribution of nutrients from mid depth to surface level waters
- Identify patterns between the clusters of nutrients, concentrations of chlorophyll, and species of phytoplankton present
- Tie in any connection between the nutrient profiles to chlorophyll’s depth profile

Data: https://simonscmap.com/catalog/datasets/HOT_Bottle_ALOHA

Our Progress...



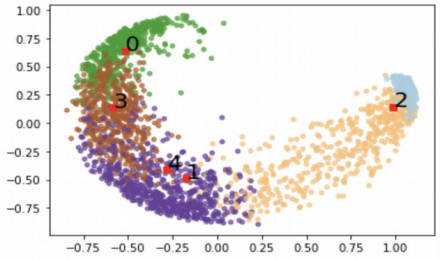
1. Data Pre-processing
    a. Subset
    b. Nutrients vs. Depth
2. Dimension Reduction [PCA]
3. Clustering Analysis [Gaussian Mixture Model / KMeans]
4. Data Visualization
    a. T-SNE (T-distributed Stochastic Neighbor Embedding)
    b. Chlorophyll vs. cluster labels
5. Supervised Learning

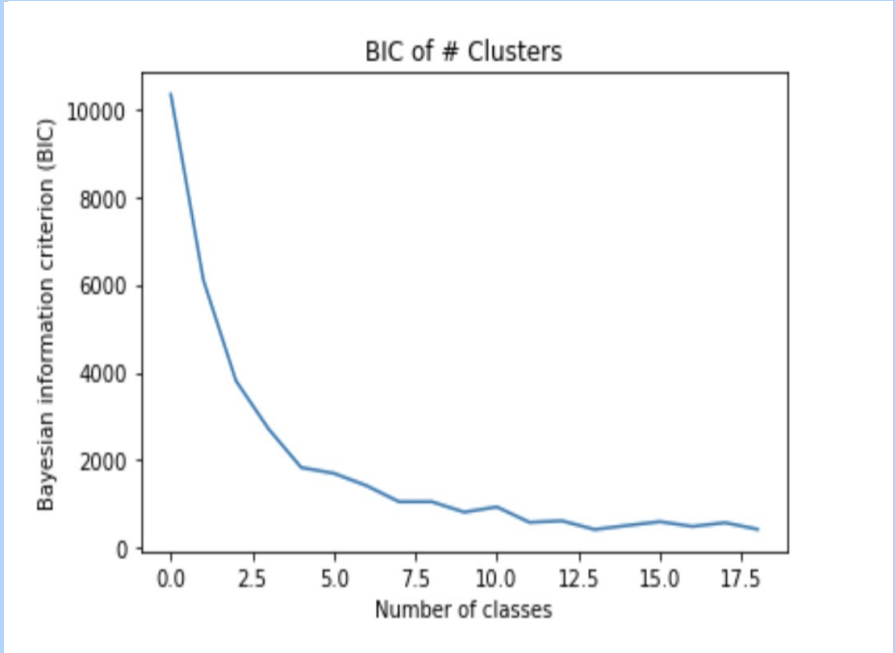


Our Progress...

3. Clustering Analysis [Gaussian Mixture Model / KMeans]

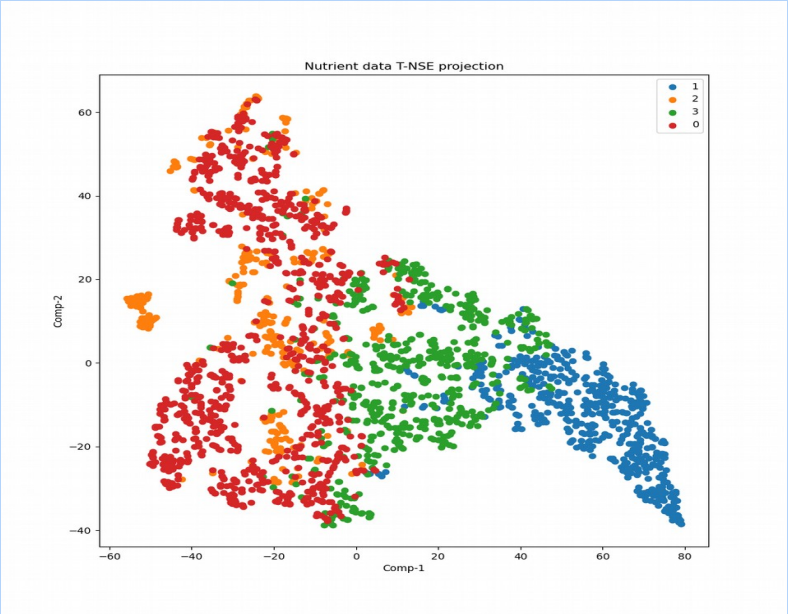
```
gmm = GaussianMixture(n_components = 5)
labels_gmm=gmm.fit_predict(X_principal)
gmm.fit(X_principal)

x_mean,y_mean=gmm.means_[:,0],gmm.means_[:,1]

# Visualizing the clustering, only use the first two components.
plt.scatter(X_principal['P1'], X_principal['P2'],
            c = gmm.fit_predict(X_principal), cmap = 'Paired', alpha = 0.7, s = 15)

for k in range(5):
    plt.plot(x_mean[k],y_mean[k],'rs',markersize=6)
    plt.annotate(str(k), (x_mean[k],y_mean[k]), fontsize=20)

plt.show()
```

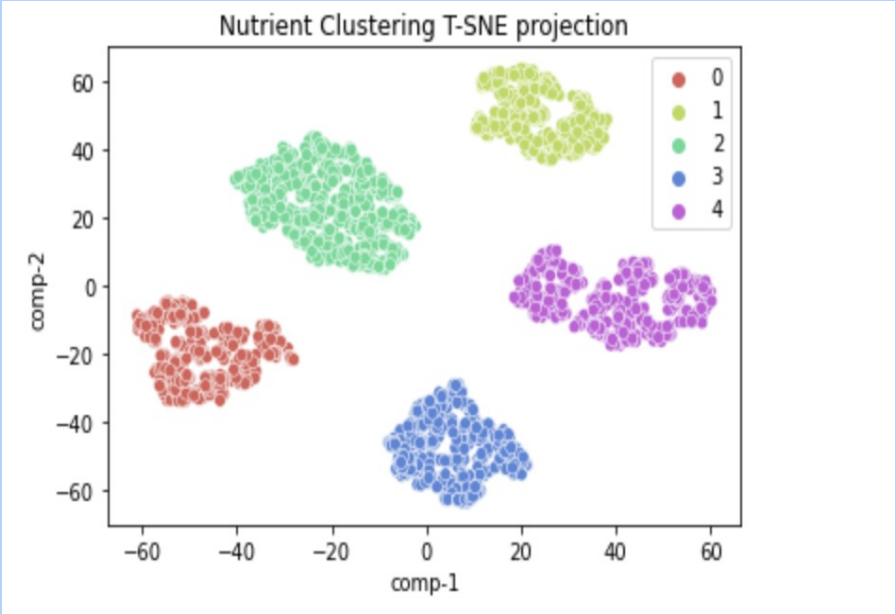




Comparing GMM results

Visualizing high dimensional data in two dimensions using **tSNE** (T-distributed Stochastic Neighbourhood Embedding)





From “Raw” Data

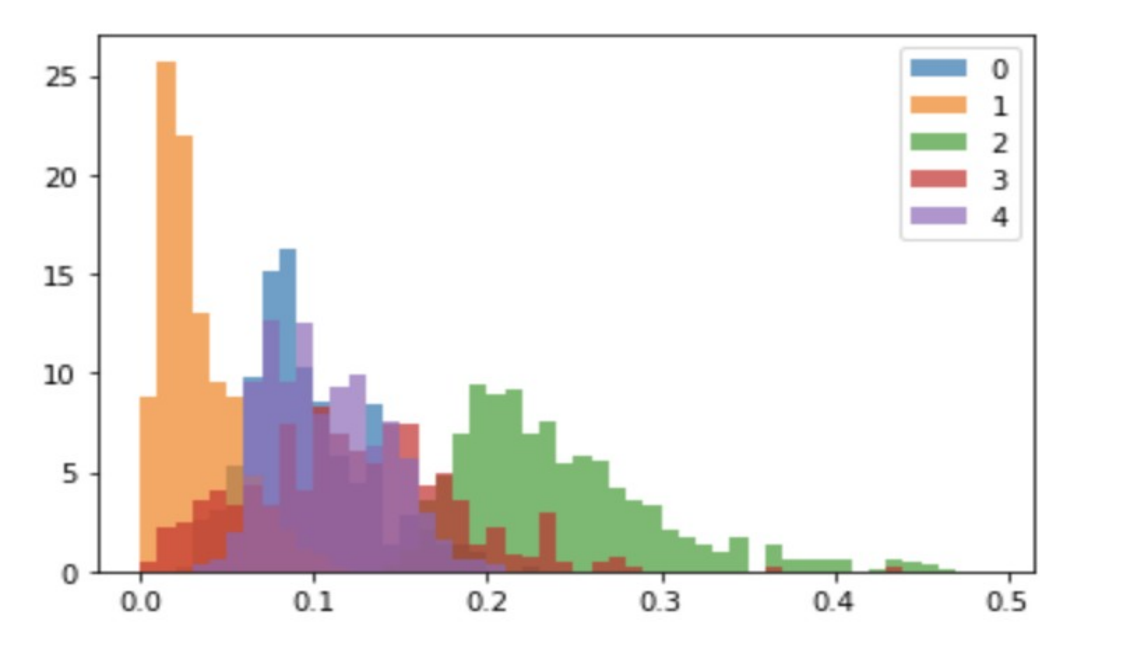
After PCA

Our Progress...

4. Data Visualization

   a. T-SNE (T-distributed Stochastic Neighbor Embedding)

   b. Chlorophyll vs. cluster labels



Our Progress...

5. Supervised Learning

Linear Regression

```
[86]: np.std(hot['chl_bottle_hot'])

[86]: 0.08542253201079253

      ...

[85]: regr = LinearRegression()
      regr.fit(X_train_pca, y_train_pca)

      pred = regr.predict(X_test_pca)
      pred_rmse = np.sqrt(mean_squared_error(y_test_pca, pred))
      print("Root Mean Square Error: " + str(pred_rmse))

      Root Mean Square Error: 0.02627582786280603
```

```
mlp_reg = MLPRegressor(hidden_layer_sizes=(150,100,50),
                       max_iter = 300,activation = 'relu',
                       solver = 'adam')

mlp_reg.fit(x_train_scaled, trainY)
```

MLPRegressor

```
MLPRegressor(hidden_layer_sizes=(150, 100, 50), max_iter=300)
```

```
print('Mean Absolute Error:', metrics.mean_absolute_error(testY, y_pred))
print('Mean Squared Error:', metrics.mean_squared_error(testY, y_pred))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(testY, y_pred)))
```

```
Mean Absolute Error: 0.016733370914193427
Mean Squared Error: 0.0005918628877406029
Root Mean Squared Error: 0.024328232318452627
```

Takeaways

What new things did we learn?

- Biological oceanography knowledge from Bottle Data of Hawaii Ocean Time-series (HOT)
- Machine learning algorithms implementation
- Learning the meaning behind TSNE, how to push a Jupyter notebook onto Github!

What was most challenging/rewarding about working on the project?

- Task workflow
- Unfamiliarity with machine learning, the significance behind PCA, and other data analysis tools used
- Team collaboration

What aspects will we keep working on?

- Find biological species that correlated with the identified nutrient clusters

Thank you!

Questions? Comments?