City of San Diego Ocean Monitoring Program | Environmental Monitoring & Technical Services Division

Making Shiny Seaworthy:

A weighted smoothing model for validating oceanographic data at sea

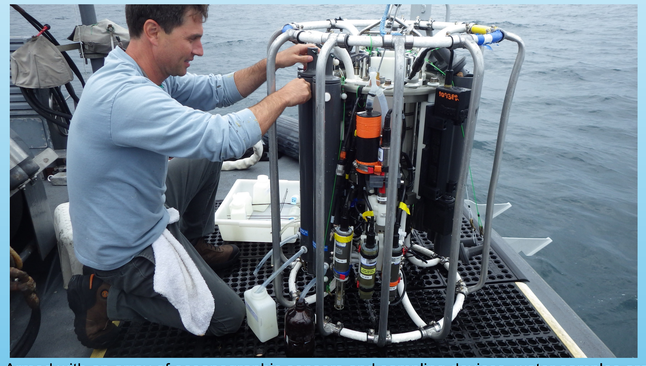
Kevin W. Byron and Matthew L. Nelson

Introduction

The City of San Diego conducts one of the largest ocean monitoring programs in the world, covering ~340 square miles of coastal waters and sampling at sea ~150 days each year. Water quality monitoring is a cornerstone of the program and requires the use of sophisticated instrumentation to measure a suite of oceanographic parameters (e.g., temperature, depth, salinity, dissolved oxygen, pH). The various sensors or probes can be episodically temperamental, and oceanographic data can be inherently non-linear, especially within stratifications (i.e., where the water properties change rapidly with small changes in depth). This makes it difficult to distinguish between extreme observations due to natural events (anomalous data) and those due to instrumentation error (erroneous data), thus, requiring manual data validation at sea.

This Shiny app improves the manual validation process by providing a smoothing model to flag erroneous data points while including anomalous data. Standard smoothing models were unable to model stratification without including erroneous data, so we elected to use a custom weighted average model where observations with a greater deviation from the local mean have less weight.

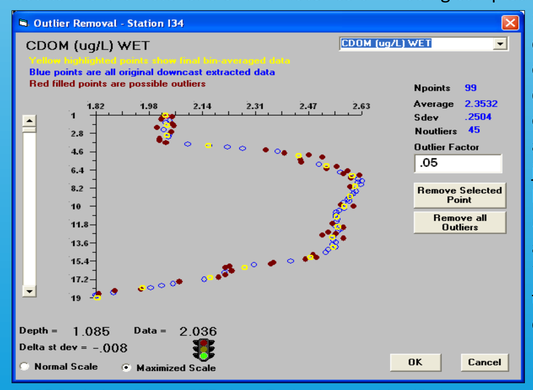
We coupled this model with an interactive Shiny session using ggplot2 and R Portable to create an offline web application for use at sea. This Shiny app takes in a raw data file, presents a series of interactive graphs for removing/restoring potentially erroneous data, and exports a new data file. Additional customization of the Shiny interface using the shinyBS package, Javascript, and HTML improve the user experience.



Armed with an array of oceanographic sensors and sampling devices, water samples are collected from the CTD while the data is processed below deck.

Current Processing of CTD Data

CTD sensors can be temperamental, so there is a need to validate the data visually to ensure results are characteristic of the water column being sampled. Large, non-linear changes in the metrics are common, if not expected; however, sudden extreme variation within minute depth changes are indicative of erroneous data, often caused by air bubbles, organic matter, or jostling of the gear. While IGODS, the current software used by The City of San Diego has a tool for addressing this issue, its algorithm is a 'black box' and tends to identify a large number of false-positives. It is also no longer commercially supported. It is our goal to replace this functionality with a reproducible tool via a shiny web interface.



IGODS window QA-ing CTD data. Red points are Identified as outliers. Operator can click on points to exclude/include. All points are retained unless explicitly removed by the biologist.

A Variation-Weighted Smoothing Model

Weighting is by the value's distance from a mean in the response's domain.

$$\hat{y}_n = [w_1 * y_1 \quad \cdots \quad w_n * y_n \quad \cdots \quad w_l * y_l]$$

$$w_n = \frac{1 - \frac{|y_n - \bar{y}|}{\sum_1^l |y_i - \bar{y}|}}{\sum_1^l \left(1 - \frac{|y_n - \bar{y}|}{\sum_1^l |y_i - \bar{y}|}\right)}$$

y-hat = estimated value for a corresponding depth, n
w = weight for the weighted average calculation
l = length of the scanning window in the smoothing model

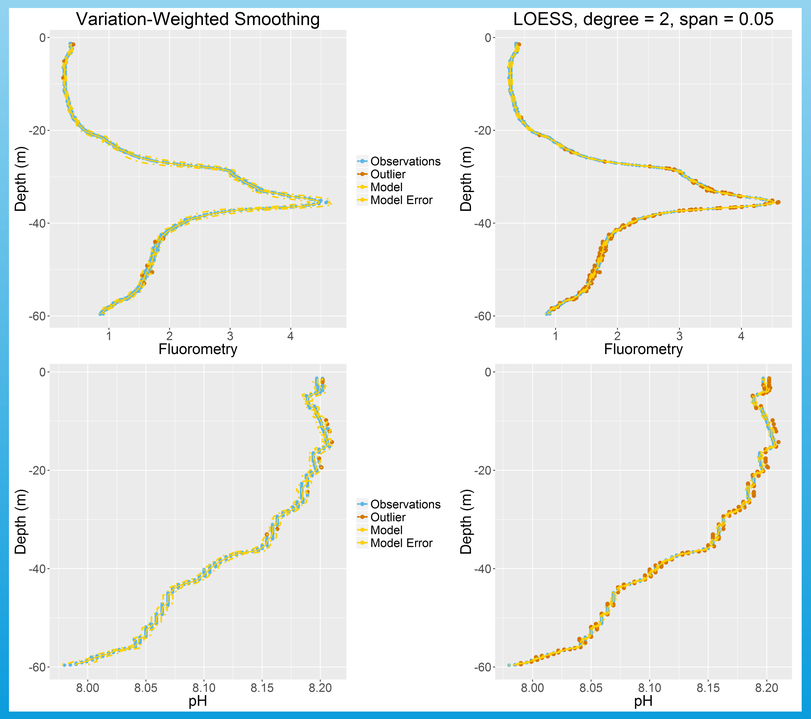
```
calcYhat <- function(df, windowSize = 9){
 for(i in length(df$y){
  ...                                                  # QC, setup, and handle window edges
  windowData <- df$y[first:last]                       # subset data for scanning window
  yBar <- mean(windowData)                             # mean value of scanning window
  ADVector <- abs(windowData - yBar)                   # Calculate Absolute Deviation
  ADcompVector <- (1 - ADvector/sum(ADvector))         # Calculate Absolute Deviation Complement
  weightVector <- ADcompVector/sum(ADcompVector)       # Convert Variation to weights

  modelData$yHat[i] <- sum(weightVector * obsWindow)   # Calculate weighted average
  modelData$ySD[i] <- sd(obs.window)                   # Calculate variation for window
 }
}
```

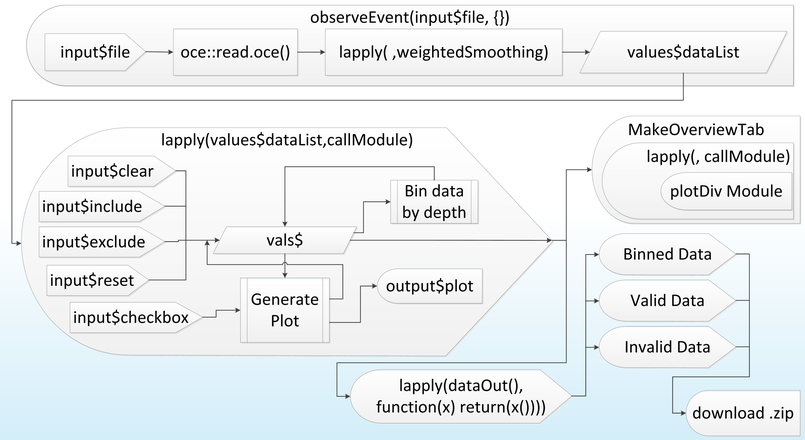
This smoothing model makes the assumption that variation between two observations is independent of the physical distance between those points. Overall variation within a window should be minimal, centered on the sensitivity of the sensor, so large variation of a single observation can be treated as an outlier. These points are identified by assigning a weight proportional to the absolute deviation from the mean of an observation.

Variation-Weighted vs. Standard Smoothing Models

Traditional smoothing techniques, like a LOESS model, had difficulties capturing environmental signals without overfitting. Below we compare a LOESS model vs. the variation-weighted smoothing applied to Fluorometry and pH data.



App Logic



This shiny app utilizes modules to create a dynamic user interface depending on what sensors are installed onto the CTD. The oce package parses seabird data files into a R-friendly list, which is passed to the smoothing function to identify erroneous data points. A module is applied over the list to generate an interactive plot for editing the data for each sensor, which creates a new reactive object for the overview tab and exporting the data.

Dependencies: shiny, shinyjs, shinyBS, DT, ggplot2, stringr, oce

Modules Create Dynamic Tabs to Adapt to Changing Data

Each sensor tab is generated from a module that returns a reactive object. The overview tab is generated with nested modules. The outer module defines the user interface, while the inner module generates each plot div. Javascript incorporates keyboard input to facilitate cycling through each tab and opening the export data modal window.

SD Public Utilities