Estimating an in situ light environment at MVCO

April 6, 2018

1 Overview

Our goal is to estimate the light intensity (and possible quality) at 4 m depth at MVCO to be able to relate division rates of *Synechococcus* to the in situ light environment (rather than the incident light) as best we can. This is involves quite a few different measurements and quite a few different assumptions for an estimation. The pieces and how they link together are diagrammed as follows:

2 Radiometer data processing

2.1 raw files to text readables

The files that are generated from the HyperPro radiometer have a ' .raw' extension. The ' .raw' files contain all the unconverted measurements from all the sensors incorporated into the HyperPro. For this exercise though, we're really only concerned with the measurements from the MPR (depth sensor), 284 (downwelling irradiance), and 285 (solar reference). Pitch, angle, roll, upwelling irradiance, and more are also measured by the HyperPro, but for just a first glance at the ligth field at depth, these can be excluded. To convert the raw files into a readable text file, we need the calibration data for each sensor and the program SatCon, which applies the conversion from the calibration files. This can all be done in with the matlab script: `processPROII_KRHC.m`. The program SatCon (as long as available in the path) is called directly from within Matlab. This script does the conversion to a text file output, and then imports the textfile to make matlab files with useful raw and processed variables. The raw files can be found in : `\\sosiknas1\lab_data\mvco\HyperPro_Radiometer\raw_data\`. In this folder, each day's measurements and casts are contained in a folder labelled by the date. The processed .txt and .mat files are stored in a similar file structure to the one found in `raw_data`, where each day has it's own folder at `\\sosiknas1\lab_data\mvco\HyperPro_Radiometer\processed_radiometer_files`. In each day folder there are `converted_txtfiles` and `mat_outfiles` folders, the latter contains .mat files for each individual casts as well as .mat files that have data and products for all the casts for that day.

With each of the light sensors there are dark measurements - these are necessary because temperature of the sensor can affect the measurement and these are corrected with corresponding dark measurements. In the `processPROII_KRHC.m` script, the nearest dark measurement in time is simply subtracted from the light measurement. PAR is calculated as the integral over wavelengths 400 - 700 nm. The light measurements are then time-synced to the MPR sensor, which has more frequent measurements. Some plots for sanity-checks are also produced if the plotflag is changed to one. If it hasn't been created, the script will prompt the user for a quality comment on each cast, such as 'not a cast' or 'good cast'. This information is used later down the line for screening of files to use in calculation of attenuation coefficients.

The following variables in the data .mat variable for each cast are:

- file name
- cruiseID (not always entered)
- operator (not always entered)
- latitude (not always entered)
- longitude (not always entered)
- timestamp
- pressure_tare
- emptyflag: data file was empty
- adj_esl: dark adjusted solar standard
- adj_edl: dark adjusted downwelling
- esl_PAR: solar standard PAR
- edl_PAR: downwelling PAR
- edl_ind: index that matches MPR data
- esl_ind: index that matches MPR data
- mprtime: matlab date
- solarflag: anything fishy for how light looks?
- depth
- wavelen_solarst: wavelengths corresponding to solarstd measurements
- wavelen_downwell: wavelengths corresponding to downwelling measurements

2.2 finding the lat/lon

Unfortunately, for a lot of the casts, the position was not entered or recorded. So, this means we have to check it against the event log and sort out which recorded lat/lon goes with each cast. Not too terrible, as there will be typically be one cast per station and the timing of the cast plus depth helps to discern position. All of this is accomplished in latlon_processing.m. It also corrects for some suspecting UTC offsets due to daylight savings time. Below is a plot of the recovered / best guesses for lat and lon. The circles represent the known stations that were visited in past all-day long cruises at MVCO, which includes the tower and the node.

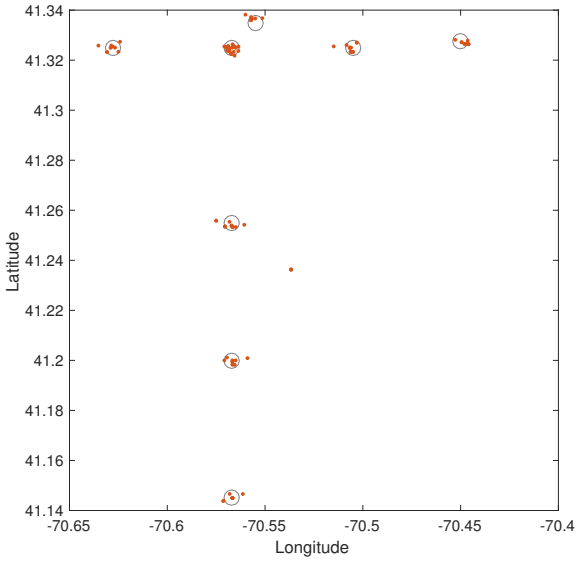

Figure 1: Recorded latitude and longitude of each cast. Records could be from either the raw datafile or the corresponding MVCO event log. If there was a discrepancy between these two, typically the MVCO event log was favored.

2.3 estimating the attenuation coefficients, k(λ) and k_{PAR}

Saved matlab variables (generated from `processPROII_KRHC.m`) are imported with the script `PAR_attenuation_coeffiecient_processing.m`, and used to estimate a the attenuation coefficient k for PAR. Now k can be estimated for each wavelength, but this is done in a later script (see below). K is calculated as the slope of a regression line that is fitted through log transformed data against depth of either light recorded at each wavelength or from PAR. Smoothed depth data from the cast is used, and measurements near the top and bottom are not used. On occasion, the points had to be manually chosen to exclude bad data from a cast. The fit, values, indices and any flags are stored in a structure, K_PAR, for each cast and save by date. This same script allows the user to load in the calculated K's and examine the fits and data points used. For some casts, a single linear regression did not seem appropriate, so the cast was split at depth (by eye) and the two pieces fit separately. A designation of '1' refers to the portion of the cast most near the surface. The various flags for the K_PAR variable and data are:

- 0: good cast
- 1: empty file or not a cast
- 2: too short a cast to get reliable k
- 3: split cast; expect two regressions

An aside note: it turns out that the ProSoft software made to do these types of calculations actually does not calculate an attenuation coefficient for PAR (it does so for each individual wavelength, and calculates PAR, but will NOT calculate K-PAR).

This data is then used to generate an attenuation coefficient for each k

2.4 Relationships with k

The script `k_relationships.m` gives some nice overview plots of where we have data for, which data we're using near the tower, and what k looks like over time. This script also imports chlorophyll data, matches k-values by date and then tries to make some relationships. Overall, things aren't terrible between k and chl, although this relationship does differ from Morel 1988 or Morel & Maritorena 2001.

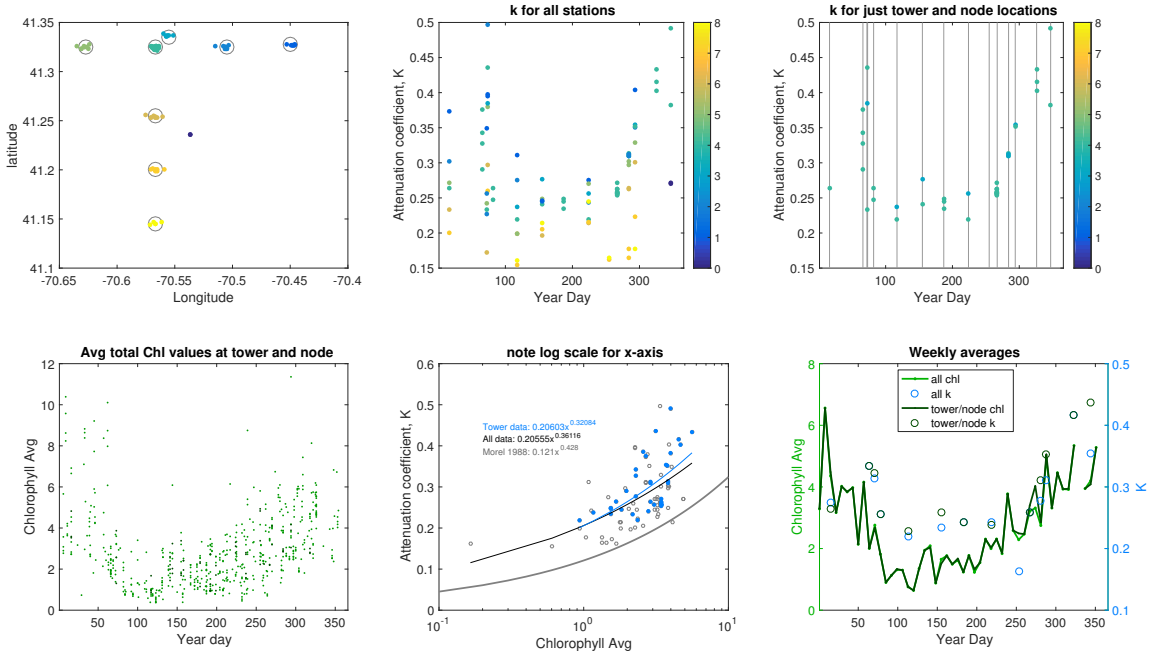

Figure 2: Plots generated from **k_relationships.m**. A) Position of all available and useable radiometer casts. B) K for PAR calculated. C) K PAR for just the node and tower. D). Average chlorophyll (mg/m3) over year day (chlorophyll averaged over all depths). E) Relationship between calculated K-PAR and average chlorophyll, with fitted power curves. F) Weekly climatologies of average chlorophyll and K-PAR values.

Another script `wavelength.m` looks at the color of light as it changes. For the most part, the light is indeed green, but the max wavelength reaching depths does shuffle about:

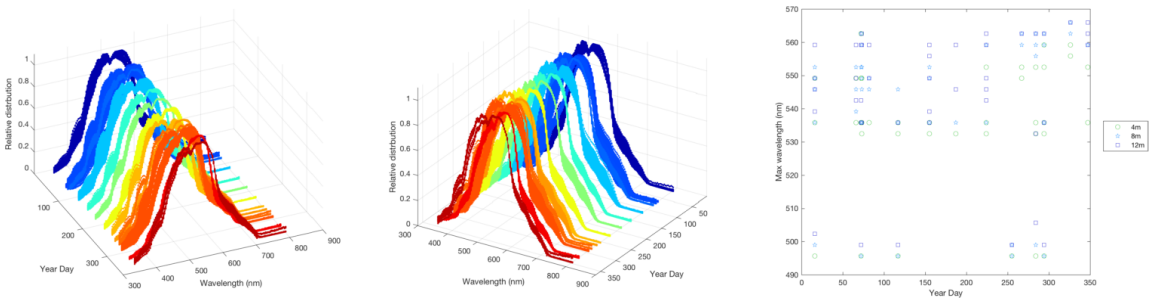

Figure 3: A) and B) showing wavelength profiles at 4m just from different views. Color coded by yearday. C) Wavelength at depth with maximum intensity.

2.5 chl-to-fluorescence match

So, there seems to be a decent relationship between chl and k, but we only have chl values for specific dates. One way around this is to use fluorometer data as a proxy for chl and make a model from that to use in a yearday k-model

3 Estimating a Mixed Layer Depth

So, now that we may have a rough idea of the light level at certain times of year, we can add another layer of information to guide our estimate. And that is trying to gauge how cells might be mixing within the water column. Typically, MVCO is well mixed, being so shallow, but stratification does happen, particularly in the summer months, and we'd like to be able to say if cells are seeing all of the water column (and light levels) or just part of it (and higher or lower light levels). Density data requires temperature, salinity and depth to calculate, but we would only have some of these variables at two depths (4m at beam, 12m at node), making it difficult to estimate what the entire water column would look like. Furthermore, salinity data at some time points is unreliable, making density estimates unavailable.

One idea is to use temperature difference between the 4m tower beam and 12m as a proxy for density difference. To see if this is a viable idea, we can examine CTD casts: look to see if they have any stratification, and how does this relate to a density difference and then temperature difference at t4 and 12 meters?

3.1 processing CTD casts

The raw CTD casts are located in a folder at: `\\maddie\TaylorF\from_Samwise\data\MVCO\`, and in order to keep processing consistent, these raw casts are processed with matlab scripts, rather than using the Seabird software (where it might be ambiguous as to what averaging / quality control is happening). The scripts that do the processing are as follows:

- `ctd_raw2cnv_processing.m`: script that process raw CTD data (either .hex or .dat files) into readable .cnv files via Sea-Bird SBE Data (note: script should be run on a machine that has this software installed).
- `import_ctd_casts.m`: imports processed CTD data (.cnv files) into a structure storage array (calls `import_cnv.m`)
- `import_cnv.m`: imports .cnv file into matlab variables

The .cnv (text readable) files have been temporarily stored in: `\\sosiknas1\Lab\data\MVCO\processed_CTD_casts\`

The data at this point is still raw, and needs further quality control. This is down with the script `CTD_QC.m`, which imports the data structure, flags bad casts, searches for downcast portion of good casts, averages downcast data over 0.2m bins, and allows user to manually remove bad data points. Quality controlled data is stored as `MVCO_QC_CTD_data.mat`.

The next step is to identify casts as either well mixed or have some evidence of stratification, suggesting a barrier to mixing. At first, I thought I could do this using the Brunt-Vaisala frequency as an indicator, but this seems to take into account the location gradients, that may or may not be indicative of true layers. (Maybe this works better for deeper water columns or coarser CTD resolution? Not sure...) At any rate, Al Pludderman (from PO department) pointed me to some well-established metrics of defining a mixed layer based on changes in density or temperature from a surface reference value. In short, the way it works is choose a surface reference value that you trust (CTD data within the first few meters can be messy), and from this value, find the depth at which density or temperature has increased or decreased past a threshold value. Some papers that look into this are Kara et al. (2000) and Brainerd and Gregg (1995), the latter which looks into how mixed layer depth metrics actually correspond to mixing.

Turns out that finding the depth at which density crosses a fixed threshold (δ) from a surface reference does a pretty good job of identifying a mixed layer. Threshold values can vary, based on season or location, but for MVCO, around $\delta 0.2$ seems to do a good job. The casts taken around MVCO seem to fall into one of three types:

- well mixed: water column is homogeneous

- surface level: strong stratification at the surface (typically < 3 or 4 m) either from fresh water or daily warming, with layer underneath well mixed and uniform
- mid-layers: mixed layer from surface (or under surface layer) until pycnocline appears at middle depths
- stratification through-out: whole water column is a pycnocline!