

Estimating an in situ light environment at MVCO

April 12, 2019

1 Overview

Our goal is to estimate the light intensity (and possible quality) at 4 m depth at MVCO to relate division rates of *Synechococcus* to the in situ light environment (rather than the incident light) as best we can. This involves quite a few different measurements and quite a few different assumptions for an estimation. At the moment, we're construction an estimate from:

- time series of incident solar radiation
- attenuation coefficients from available radiometer
- estimation of stratification at MVCO from continuous temperature records at two depths.

2 Radiometer data processing

The scripts used to do so are summarized below:

script	description	specific products
<code>processPROII_MVCO.m</code>	converts raw files to matlab and text files, corrects with dark measurements and calculates PAR	see Table 2
<code>latlon_processing.m</code>	identifies latitude and longitude for all casts	
<code>PAR_attenuation_coeffiecient_processing.m</code>	calculate and save k_{PAR}	
<code>wavelength_attenuation_coefficient_processing.m</code>	calculate and save $k(\lambda)$'s	
<code>k_relationships.m</code>	organizes k-values for PAR, overview plots, explores relationships with chl	<code>k_lite.mat</code>
<code>wavelength.m</code>	exploration and plots of spectra over yearday	
<code>apply_k_to_incident_light.m</code>	interpolate k values for annual cycle and apply to incident radiation	<code>k_interp.mat</code>

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 our purposes, 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 those will be excluded for now. 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 matlab with the script: `processPROII_MVCO.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.

For each of the light sensors there are corresponding dark measurements - these are necessary because temperature of the sensor can affect the measurement. This is corrected using the dark measurements. In `processPROII_MVCO.m`, 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.

variable	info
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. The lat/lon for each cast is recorded in structure variable in a separate .mat file in each day's cast folder as described above.

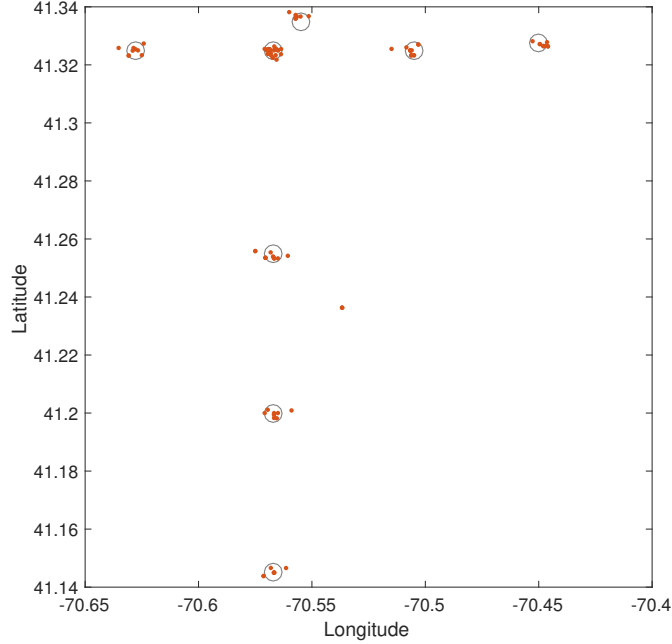


Figure 1: Recorded latitude and longitude of each cast. Records could be from either the raw datafile or the corresponding MVCO event log. If there was a discrepancy between these two, typically the MVCO event log was favored.

2.3 Estimating attenuation coefficients, $k(\lambda)$ and k_{PAR}

Saved matlab variables (generated from `processPROII_MVCO.m`) are imported with the script `PAR_attenuation_coefficient_processing.m`, and used to estimate 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

All of these variables and fits are saved in a structure variable, in a separate .mat file in each cast folder as above. A side 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).

To calculate k for each wavelength, this is done with `wavelength_attenuation_coefficient_processing.m`, which outputs a similar structure variable as for k_{PAR} .

2.4 Relationships with k

The script `k_relationships.m` aggregates and organizes the k_{PAR} values over all the casts for use downstream (i.e. to calculate climatology of light at depth). It also 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. While there may be decent relationship between chl and k , 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. This was more in-depth than we had time for, but would be cool to explore in the future!

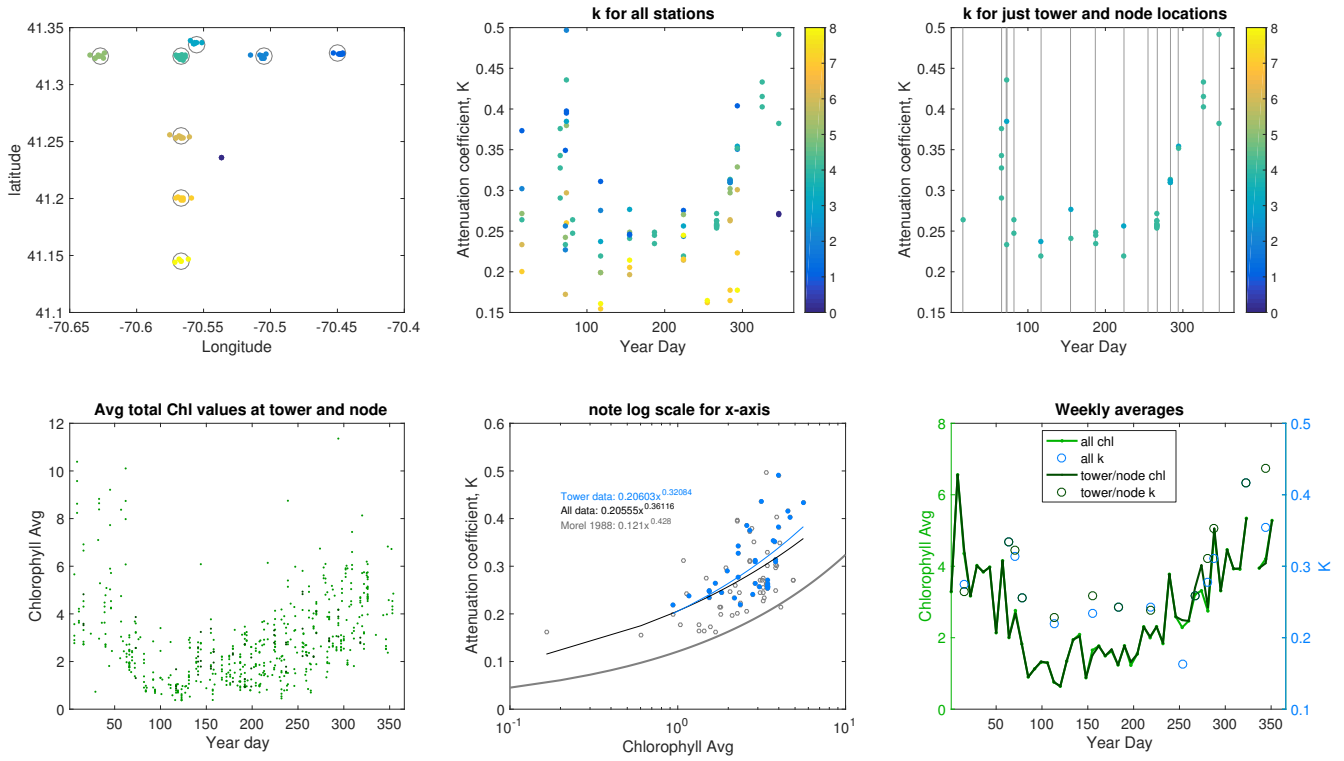


Figure 2: Plots generated from **k_relationships.m**. A) Position of all available and useable radiometer casts. B) K for PAR calculated. C) K PAR for just the node and tower. D). Average chlorophyll (mg/m³) over year day (chlorophyll averaged over all depths). E) Relationship between calculated K-PAR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

Another script `wavelength.m` looks at the distribution of energy in different wavelengths, and at different depths. For the most part, the light is indeed green, but the wavelength with the maximum energy can shift towards blues, depending on time of year.

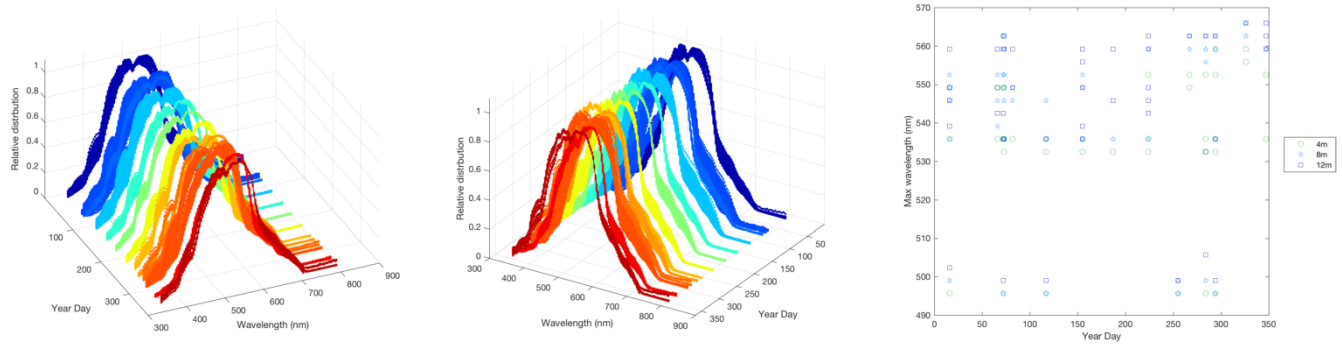


Figure 3: A) and B) showing wavelength profiles at 4m just from different views. Color coded by yearday. C) Wavelength at depth with maximum intensity.

3 Evaluating water column stratification

So, now that we 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 only have some of these variables at two depths (4m at beam, 12m at node) over time, making it difficult to estimate what the entire structure of the water column would look like. Furthermore, salinity data at some time points is unreliable, making density estimates unavailable.

One idea is to use the 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 4 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:

script	description	specific products
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	
CTD_QC.m	quality control processing of CTD casts	QC_downcast.mat
find_mixed_layer_depth.m	calculates when potential density crossed threshold values in water column	mld.mat
examine_tempdate_hourly.m	examine beam and node temperature records with stratification index	temp_hourly_beam_node.mat

The first 3 scripts are located in the /NES-LTER/ctd_processing. 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 done 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 QC_downcast.mat in \\sosiknas1\lab_data\MVCO\processed_CTD_casts\.

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 local 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 Plummer (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, choose a surface reference value that you trust (CTD data within the first few meters can be unreliable), 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 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 $\Delta 0.2 \text{ kg/m}^3$ for density seems to be it. The casts taken around MVCO seem to fall into one of four types:

- well mixed: water column is homogeneous
- surface stratification: strong stratification at the surface (typically < 3 or 4 m) either from fresh water or daily warming, with layer underneath well mixed and uniform to bottom
- 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!

Most of the casts available (104 at present, but only 72 unique days) were well mixed, and from looking at the casts taken on the same day, the water column can change quite rapidly (indicating possible daily stratification and break down, which would be normal for a site at this shallow depth). This processing was done with `find_mixed_layer_depth.m`, a messy script, which categorizes when different thresholds were crossed at which depth. This is saved in `ml.d.mat`.

The goal though is to evaluate how well temperature differences can serve as proxy for density differences (which in turn says something about the water column structure). We calculate the change in temperature and density from 4m and 12m in the cast and see how well this relates to the classification of the water column (which used the whole profile). In general, this is not bad (although a change in density and temperature will miss cases where there is just surface stratification). We also find that, in general, a change in temperature strongly correlates to change in density, lending support to using just temperature as a proxy for density:

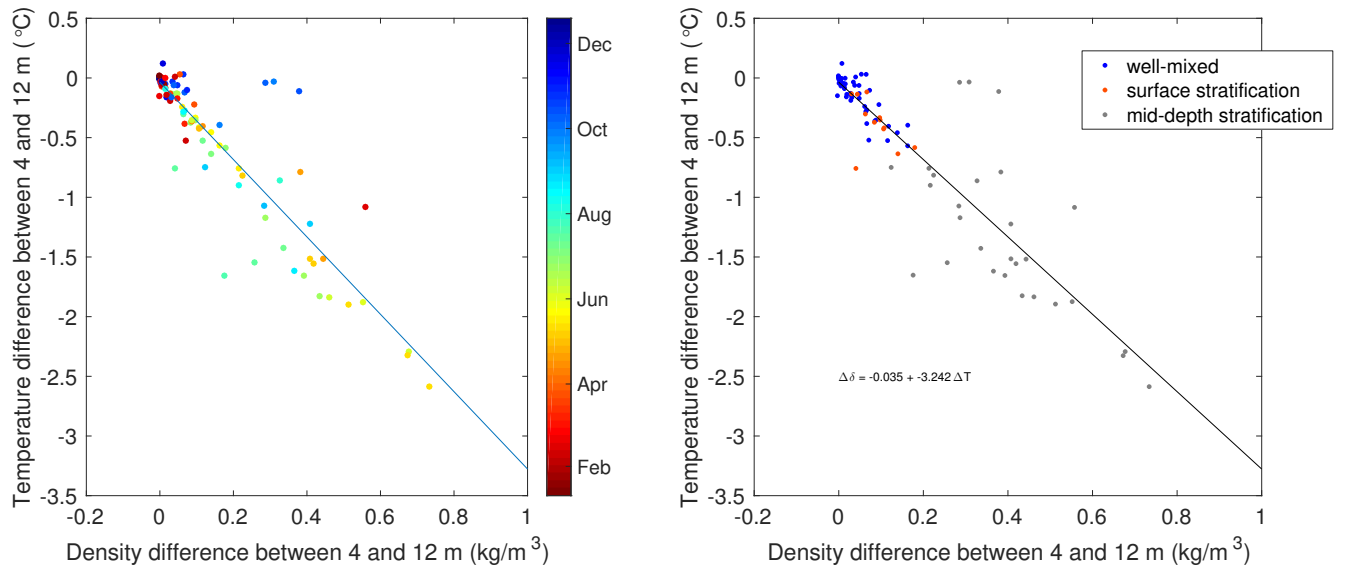


Figure 4: A) and B) shows relationship between the difference in potential density at 4 and 12 m and the difference in temperature for those two depths, recorded in the CTD cast. B) is color-coded based on stratification classification.

A $\Delta 0.2$ in density corresponds to around $\sim 0.68^\circ\text{C}$ temperature change (based on the linear regression curve in Fig. 4B, although, one could calculate this based on equation of state of seawater...). Using this as a cutoff, we can now look over the entire timeseries of beam (4 m) and node (12 m) temperature records to get an idea of seasonal stratification! Woo!

Looking at the daily averaged temperature records for the beam and node is not enough, as this average can be misleading by aggregating over day and night. Instead, we look at hourly time points to see 1) how the water column can change over the day and 2) see how many hours of the daylight portion would be considered ‘stratified’. These actions are done with the script `examine_tempdate_hourly.m`.

It seems that the change in temperature over a day can vary quite a bit (up to a few degrees for both beam and node temperature records). The maximum difference between these two records for a given hour shows a curious seasonality, with the largest differences observed in winter and summer:

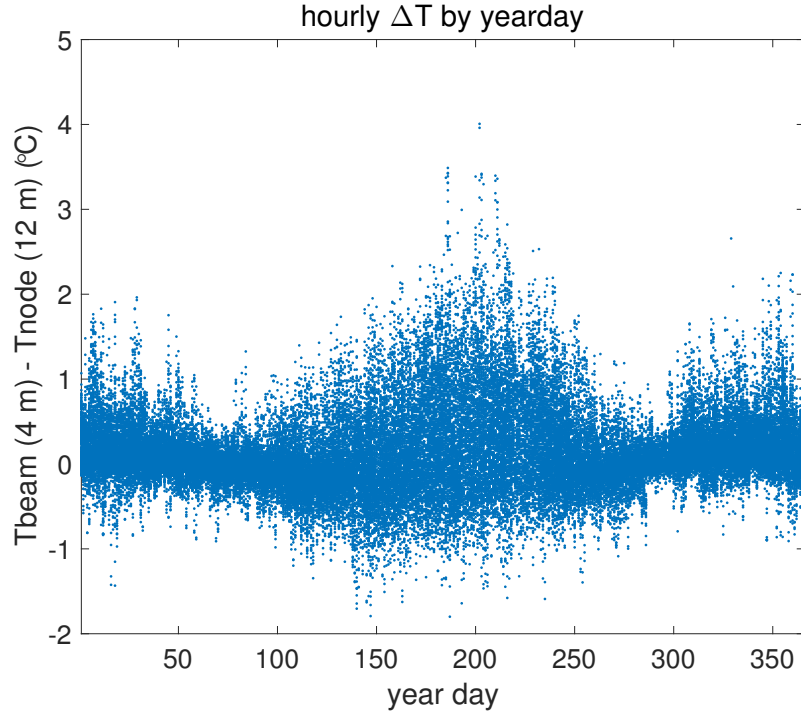


Figure 5: Hourly difference in temperature between 4 and 12 plotted against yearday for that hour.

We find that a majority of hours during a day seems to be well mixed. With the exception of days in summer and in winter, very few days would be stratified as to limit cell movement within the water column (although we still have no idea of how fast these cells would be mixing...). Figure 6 illustrates this by showing a frequency bar graph of days in the dataset that had a certain percentage of their hours ‘stratified’ as based on the $\Delta 0.68^\circ\text{C}$ proxy. A day is considered stratified if more than 2/3 of its daylight hours had a temperature difference that exceeded 0.68°C . This comes out to about %97 of the days being stratified!

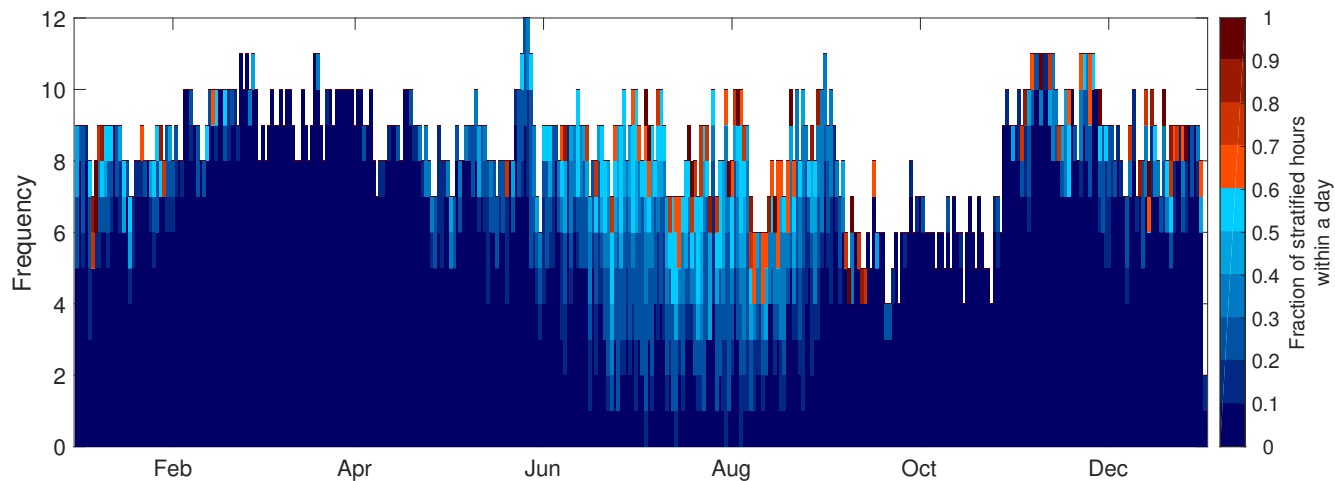


Figure 6: Bar graph showing how many days (over the entire time series) had a certain percentage of daylight hours considered ‘stratified’ based on the $\Delta 0.68^{\circ}\text{C}$ proxy. The low number at the end of the year is for leap year days.

4 Application of k to incident light

Now safely assuming that we have a well mixed water column, we can calculate an average light environment with `apply_k_to_incident_light.m` by assuming the cells pass through the entire water column in a day. Assuming simple exponential attenuation, we have:

$$\bar{E}_d(t) = \frac{1}{15} \int_{15}^0 \bar{E}_0(t) \cdot \exp(\bar{K}_d(t) \cdot z) dz, \quad (1)$$

where $\bar{E}_0(t)$ is the climatological value of incident radiation on year day t , $\bar{K}_d(t)$ is the interpolated K_d value, and $\bar{E}_d(t)$ is the resulting average light level. Here, 15 refers to the average height in meters of the water column at MVCO. This calculation can be performed analytically and comes out to:

$$\bar{E}_d(t) = \frac{\bar{E}_0(t)}{15 \cdot \bar{K}_d(t)} [1 - \exp(15 \cdot \bar{K}_d(t))] \quad (2)$$

This script saves the interpolated k 's and light at depth products in `k_interp.mat` in the repository.