RSKtools for Matlab processing RBR data

Table of Contents

RSKtools v2.0.0; RBR Ltd. Ottawa ON, Canada; support@rbr-global.com; 2017-07-07

Introduction

A suite of new functions are included in RSKtools v2.0.0 to post-process RBR logger data. Below we show how to implement some common processing steps to obtain the highest quality data possible.

RSKtools help

All post-processing functions are customisable with name-value pair input arguments. To process all data using the default parameters no name-value pair arguments are required. All the information above is available for each function using `help`, for example: `>> help RSKsmooth`.

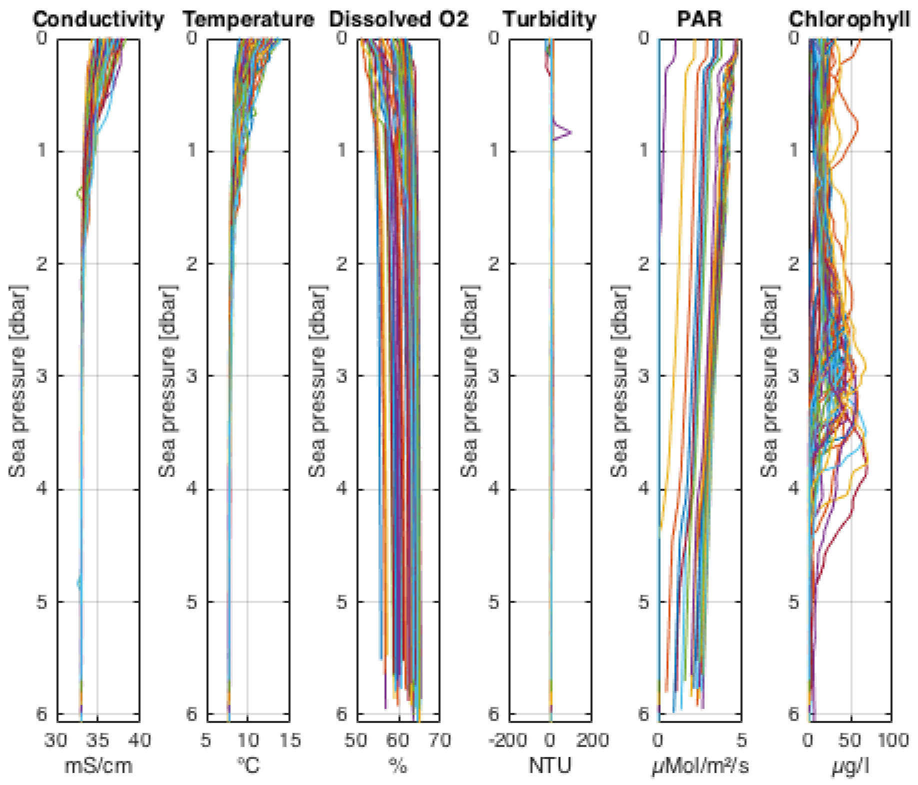
Getting set up

Review VignetteStandard for help.

```
% First, open a connection to the RSK logger database file:
file = 'sample.rsk';
rsk = RSKopen(file);

% read the upcast from profiles 1 - 55
rsk = RSKreadprofiles(rsk, 'profile', 1:55, 'direction', 'up');

% plot the raw data as profiles
RSKplotprofiles(rsk);
```



Remove atmospheric pressure from measured total pressure

We suggest deriving sea pressure first, especially when an atmospheric pressure other than the nominal value of 10.1325 dbar is used, because deriving salinity and depth requires sea pressure.

```
rsk = RSKderiveseapressure(rsk);

% Hang on to the raw data for plotting later.
raw = rsk;
```

Low-pass filtering

Applying a low pass filter to temperature and conductivity smooths high frequency variability and compensates for differences in sensor time constants (the thermistor often has a slower response to changes than conductivity). RSKtools includes a function called `RSKsmooth` for this purpose.

RBR thermistors on profiling instruments have time constants of about 0.6 s, so the conductivity should be smoothed to match that value. In this example, the logger samples at 6 Hz instrument (`rsk.continuous.samplingPeriod`), so a 7 sample window should provide sufficient smoothing.

```
rsk = RSKsmooth(rsk, {'Conductivity', 'Temperature'}, 'windowLength',
 7);
```

Alignment of conductivity to temperature and pressure

Conductivity, temperature, and pressure need to be aligned in time to account for the fact these sensors are not physically co-located on the logger. At any instant, the sensors are measuring a slightly different parcel of water. Accounting for this effect ensure that salinity is of high accuracy.

The classic approach is to compute the salinity for a range of lags, plot each curve, and to manually choose the curve with the smallest salinity spikes at sharp interfaces. As an alternative, RSKtools provides a function called `RSKcalculateCTlag` that estimates the optimal lag between conductivity and temperature by minimising salinity spiking. See `help RSKcalculateCTlag`.

```
lag = RSKcalculateCTlag(rsk);
rsk = RSKalignchannel(rsk, 'Conductivity', lag);
```

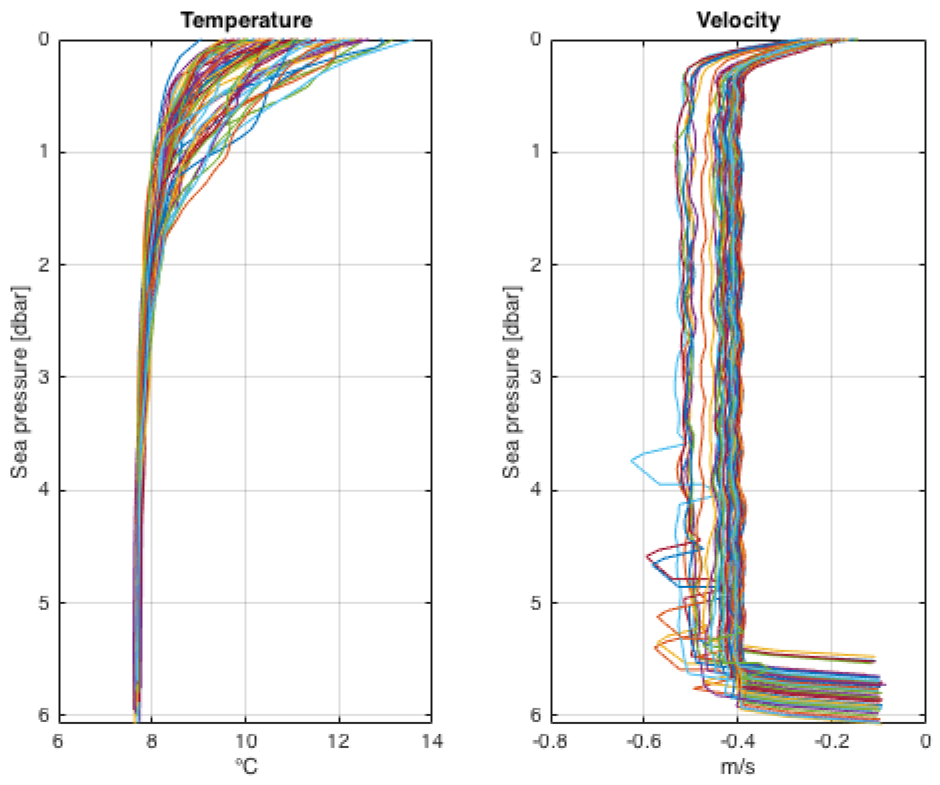
Remove loops

Profiling during rough seas can cause the CTD descent (or ascent) rate to vary, or even temporarily reverse direction. During such times, the CTD can effectively sample its own wake, potentially degrading the quality of the profile in regions of strong gradients. The measurements taken when the instrument is profiling too slowly or during a pressure reversal should not be used for further analysis. We recommend using `RSKremoveloops` to flag and remove data when the instrument falls below a `threshold` speed. Use `RSKderivedepth` to calculate depth from sea pressure, and `RSKderivevelocity` to calculate profiling rate.

```
rsk = RSKderivedepth(rsk);
rsk = RSKderivevelocity(rsk);

% Plot the velocity profile to help determine a minimum rate
RSKplotprofiles(rsk,'channel',
{'temperature','velocity'},'direction','up');

% Apply the threshold
rsk = RSKremoveloops(rsk, 'threshold', 0.3);
```



Derive practical salinity

RSKtools includes a function to derive Practical Salinity using the TEOS-10 GSW function `gsw_SP_from_C`. The TEOS-10 GSW Matlab toolbox is freely available from http://www.teos-10.org/software.htm.

```
rsk = RSKderivesalinity(rsk);
```

Bin average all channels

Average the data into 0.5 dbar bins using `RSKbinaverage`.

```
rsk = RSKbinaverage(rsk, 'binBy', 'Sea Pressure', 'binSize',
 0.5, 'direction', 'up');
```

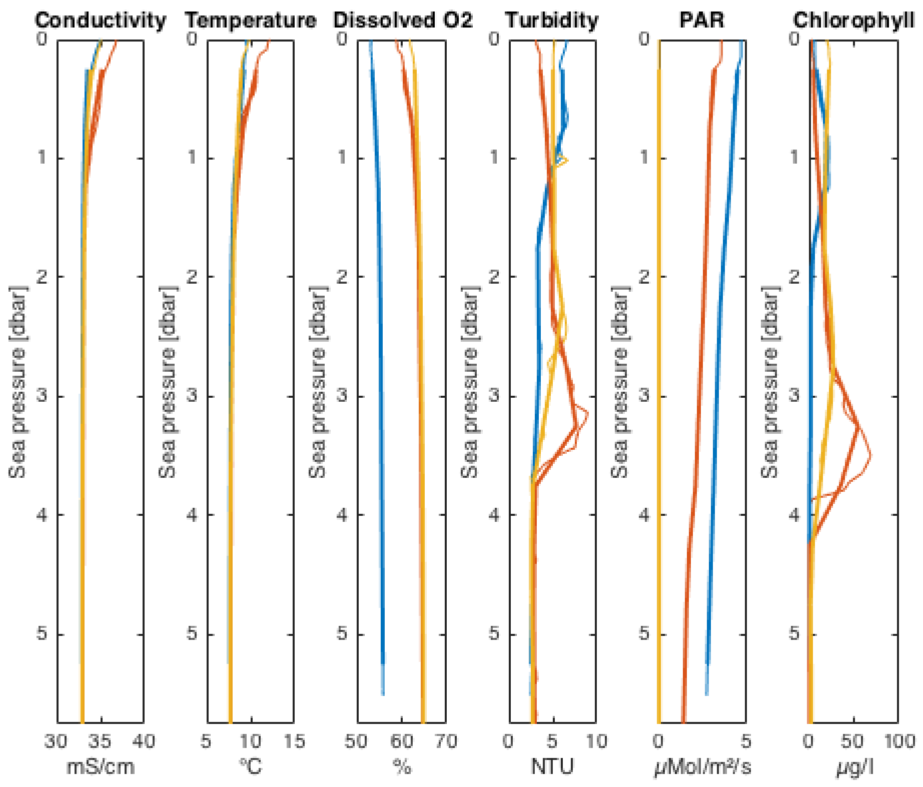
Plot the bin averaged profiles

Compare the binned data to the raw data for a few example profiles.

```
clf
RSKplotprofiles(raw,'profile',[1 20 40]);
axhandles = get(gcf,'children');
for k=1:length(axhandles),
  hold(axhandles(k),'on')
```

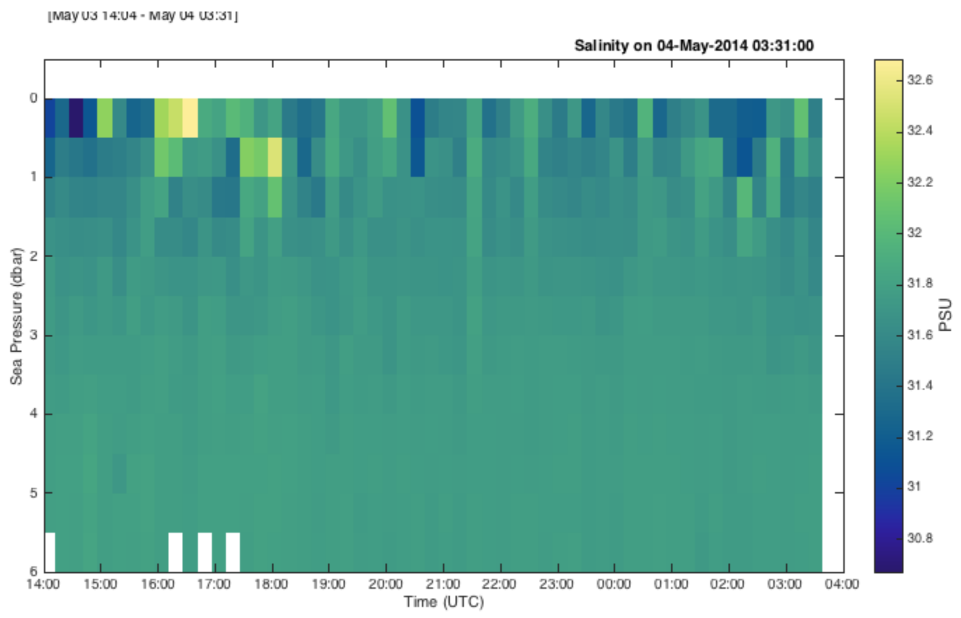
```
end

hdls = RSKplotprofiles(rsk,'profile',[1 20 40],...
                          'channel',{'conductivity','temperature',...
                          'dissolved
 O2','turbidity','par','chlorophyll'});
set(hdls,{'linewidth'},{2})
```



2D plot

```
clf
RSKplot2D(rsk, 'Salinity');
```



Display a summary of all the processing steps

Type `rsk.log{:,2}` at the command prompt.

Other Resources

VignetteStandard is available for information on getting started with `RSKtools` standard functions.

A User Manual for RSKtools is available.

About this document

This document was created using Matlab™ Markup Publishing. To publish it as an HTML page, run the command:

```
publish('RSKtools_vignette.m');
```

See `help publish` for more document export options.

Published with MATLAB® R2015b