The R oce package for oceanographic data analysis

Using oce for working with Argo data

Clark Richards[1], Dan Kelley[2], Jaimie Harbin[1]

[1] Bedford Institute of Oceanography Dartmouth, NS, Canada
[2] Dalhousie University Halifax, NS, Canada

2020-01-28

Outline

1. Introduction

2. Why R?

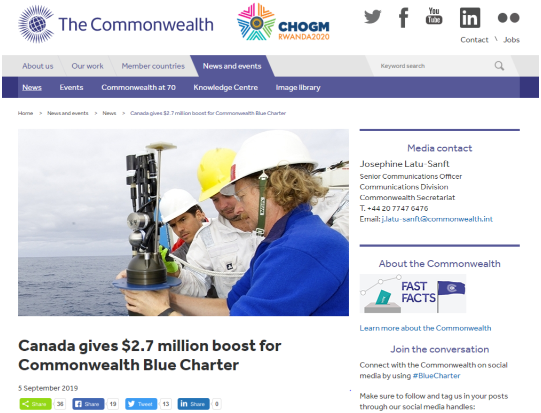
3. The oce package

4. Working with Argo data

5. The future of oce and Argo

Introduction

As part of its contribution to the Commonwealth Blue Charter, Fisheries and Oceans Canada is contributing funding for various initiatives, including advancing BGCArgo, especially for Small Island Developing States (SIDS).



Canada gives $2.7 million boost for Commonwealth Blue Charter

5 September 2019

Media contact

Josephine Latu-Sanft
Senior Communications Officer
Communications Division
Commonwealth Secretariat
T. +44 20 7747 6476
Email: j.latu-sanft@commonwealth.int

About the Commonwealth

Learn more about the Commonwealth

Join the conversation

Connect with the Commonwealth on social media by using #BlueCharter

Make sure to follow and tag us in your posts through our social media handles:

Development of accessible science tools

One component of this initiative is to support science in SIDS by developing open and accessible science tools, to allow access to data (such as Argo and BGCArgo) that has a relatively high barrier to newcomers.

The tools should be:

- Freely available
- Easily installable
- Easy to use, and
- Useful (for science/research)

Future work may also involve collaborative training and workshops to develop and share expertise with Argo tools and data.

What is R?

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS.

[1] CRAN: the Comprehensive R Archive System

What is R?

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS.

R is also:

1. *free* (as in speech and as in beer) and open-source
2. designed specifically for working with data
3. extended through a comprehensive package system[1], that is *curated* and centrally maintained. Over 15000 packages currently.
4. increasingly used in undergraduate statistics, science, and data science programs

[1] CRAN: the Comprehensive R Archive System

R packages

R packages, installed within R itself, have to adhere to strict criteria involving:

- code checks
- proper documentation
- functioning examples
- cross-platform compatibility
- package-specific unit tests (optional – to ensure that package changes don't affect specific numeric results)

This means that installed R packages "just work".

R packages

R packages, installed within R itself, have to adhere to strict criteria involving:

- code checks
- proper documentation
- functioning examples
- cross-platform compatibility
- package-specific unit tests (optional – to ensure that package changes don't affect specific numeric results)

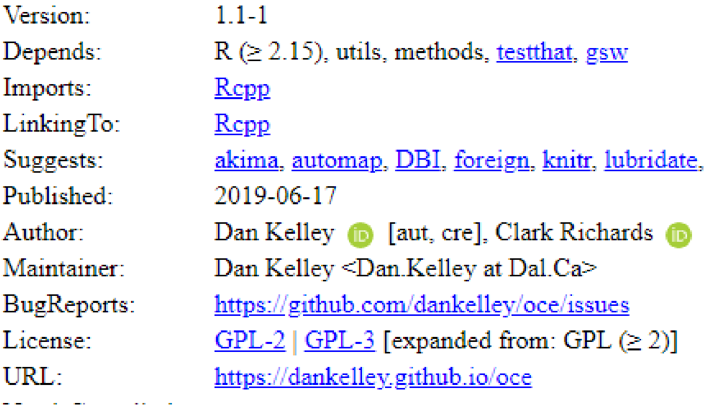
This means that installed R packages "just work".

Installing packages can be as simple as:

```r
install.packages(c("oce", "cmocean"))
```

R packages cont'd

R packages make use of *dependencies* meaning that new packages can build off of other existing packages



Version:	1.1-1
Depends:	R (≥ 2.15), utils, methods, testthat, gsw
Imports:	Rcpp
LinkingTo:	Rcpp
Suggests:	akima, automap, DBI, foreign, knitr, lubridate,
Published:	2019-06-17
Author:	Dan Kelley [aut, cre], Clark Richards
Maintainer:	Dan Kelley <Dan.Kelley at Dal.Ca>
BugReports:	https://github.com/dankelley/oce/issues
License:	GPL-2 | GPL-3 [expanded from: GPL (≥ 2)]
URL:	https://dankelley.github.io/oce

Object orientation

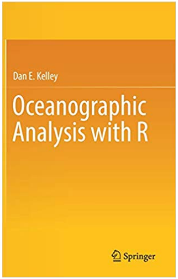
The R language includes a few different types of *object orientation*, which can be of great benefit for consistently working with specific data files or types.

Packages can define their own *classes* and *methods* to accompany those classes.

This will be demonstrated in the context of the “oce” package.

The “oce” package

pronounced “O-see-ee” for the “oce computational environment”


Dan E. Kelley

Oceanographic Analysis with R

Springer

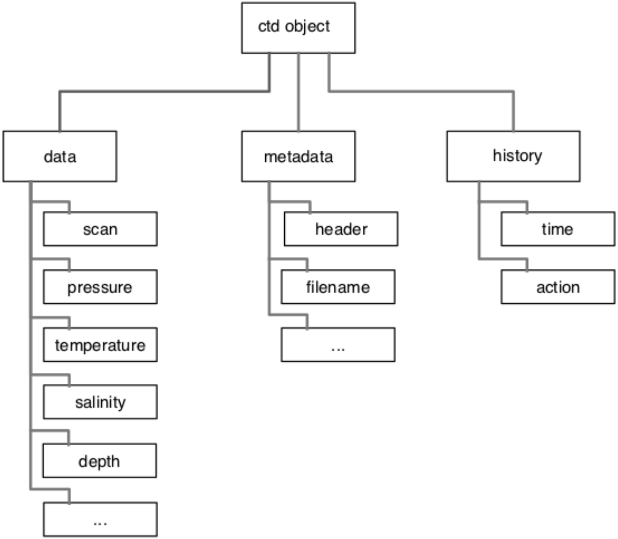
From `https://dankelley.github.io/oce/index.html`

Oce is a package for the R statistical language that helps Oceanographers do their work by providing functions to read Oceanographic data files, to process the data in instrument-specific ways, and to represent the results with plots that follow Oceanographic convention.

What can oce do?

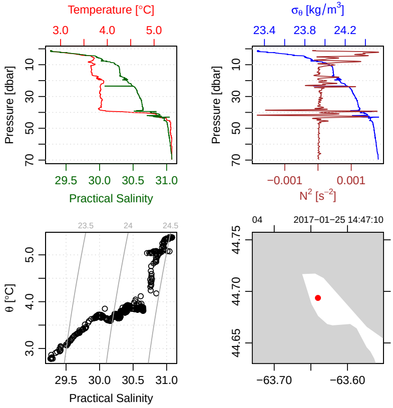
- read standard and manufacturer specific data files (SBE, RBR, WOCE, RDI, Nortek, ArgoNetCDF, landsat, AMSR, xbt, ...)
- creates a core “oce” class for storing objects
- subclasses for different object types:
  - ctd, adp, section, sealevel, topo, xbt, argo, ...
- creates generic *methods* for plotting and handling objects
- builds off the “gsw” package (TEOS-10) for equation of state
- standard CTD processing routines, e.g. `ctdTrim()`, `ctdDecimate()`, `ctdFindProfiles()`

Oce object structure



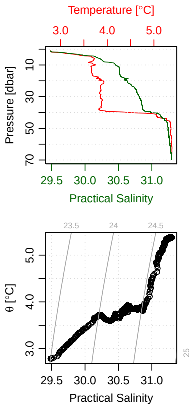
CTD example: SBE cnv

```
library(oce)
d1 <- read.oce("17667004_align_tm.cnv")
d1 <- subset(ctdTrim(d1, method = "sbe"),
    pressure < 70)
plot(d1)
```



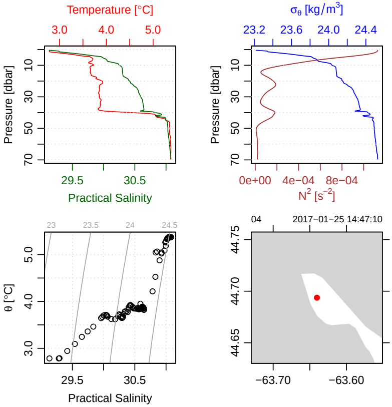
CTD example: RBR rsk

```
d2 <- as.ctd(read.oce("060346_20170127_1615.rsk"))
d2 <- subset(ctdTrim(ctdFindProfiles(d2)[[3]],
    method = "sbe"), pressure < 70)
plot(d2)
```



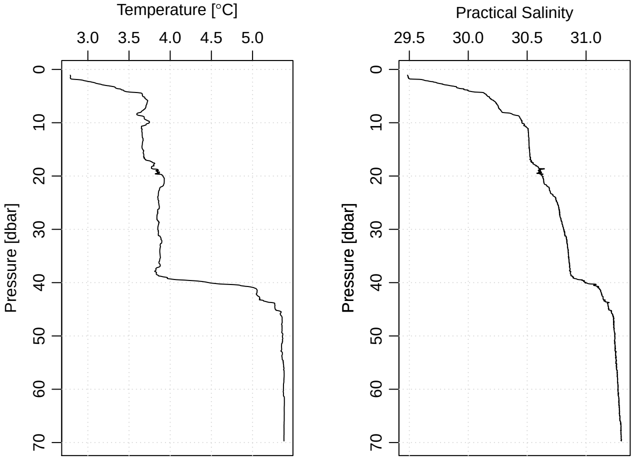
CTD example: DFO ODF (“ocean data format”)

```
d3 <- read.oce("D17667004.ODF")
plot(d3)
```



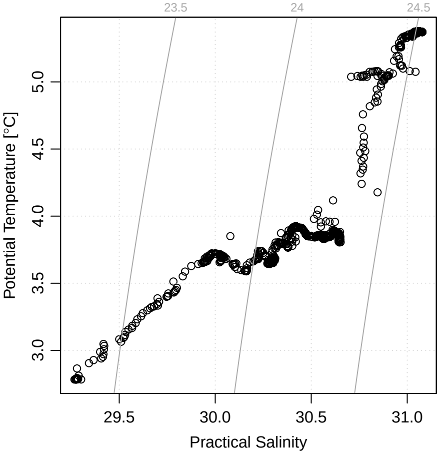
Oce-specific plots 1: profiles

```
par(mfrow = c(1, 2))  # 2 column plot
plotProfile(d2, xtype = "temperature")
plotProfile(d2, xtype = "salinity")
```



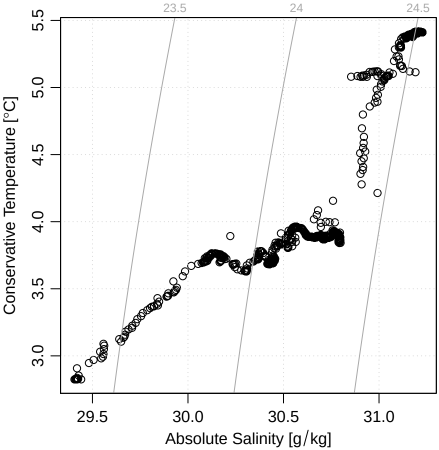
Oce-specific plots 2: TS plot

```
plotTS(d1)
```



Oce-specific plots 3: C_T vs S_A plot with TEOS-10

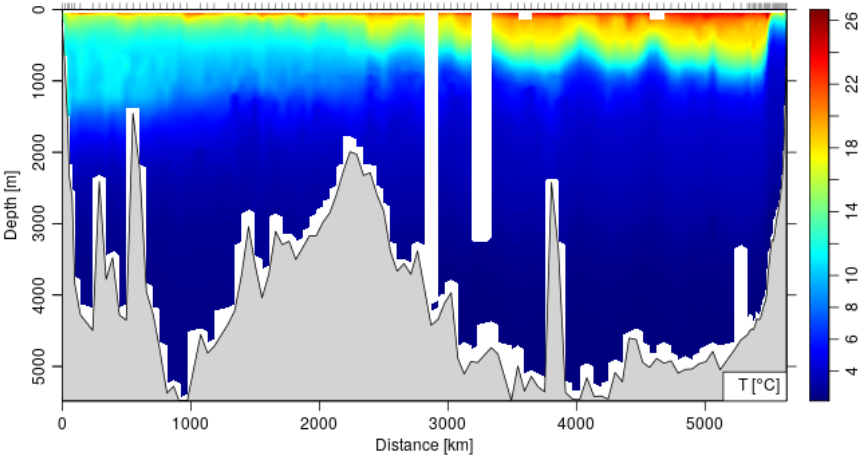
```r
plotTS(d1, eos = "gsw")
```



Working with "section" data

A “section” object is a collection of ctd objects with the relevant metadata describing station information.

```
plot(section, which = "temperature", ztype = "image")
```

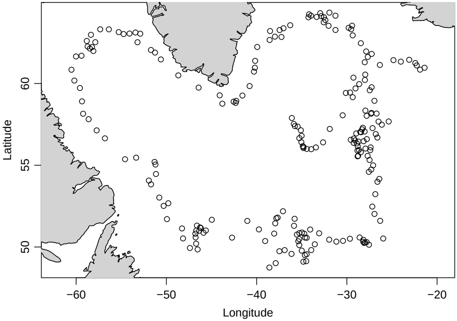


Working with Argo data

Oce can read the standard Argo netCDF files, and creates an oce object of class `argo`.

An example dataset is included when oce is installed (float 6900388):

```
data(argo)
plot(argo)
```



Handling data quality flags

Data quality flags are an important feature of many oceanographic data sets (including Argo).

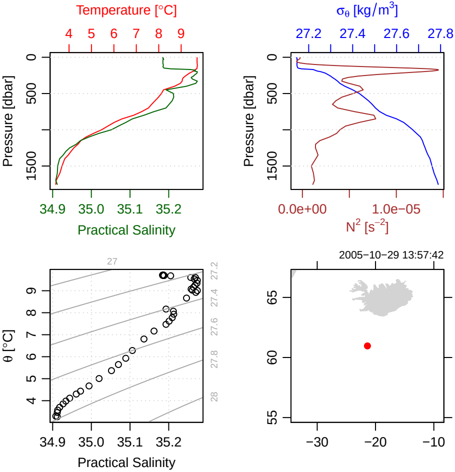
Oce contains a *generic* function, `handleFlags()` designed to work with them. `handleFlags()` has a method for argo objects:

```
a <- handleFlags(argo)  # default keeps only flag==1 v

str(argo[["flagScheme"]])  # str shows the 'structure'

## List of 2
##  $ name   : chr "argo"
##  $ mapping:List of 8
##   ..$ not_assessed     : num 0
##   ..$ passed_all_tests: num 1
##   ..$ probably_good    : num 2
##   ..$ probably_bad     : num 3
##   ..$ bad              : num 4
```

Converting between classes

Data from `argo` class objects can be converted (“coerced”, in R-speak) to other relevant oce classes, e.g.

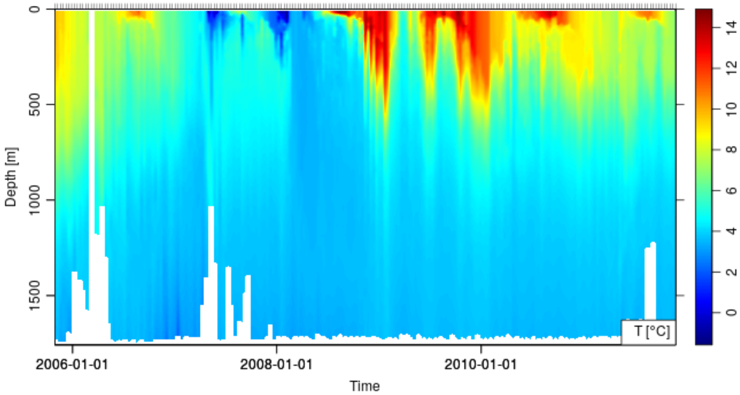
```
plot(as.ctd(a, profile = 1))
```



Converting between classes

Data from `argo` class objects can be converted ("coerced", in R-speak) to other relevant oce classes, e.g.

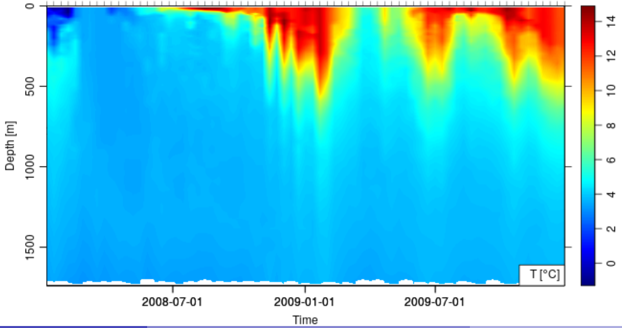
```
as <- as.section(a)
plot(as, which = "temperature", xtype = "time", ztype
= "image")
```



Subsetting data: by time

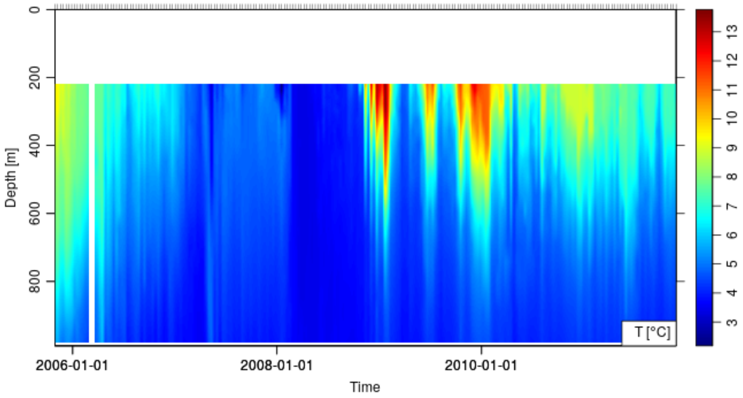
Another useful generic for argo-class objects is `subset()`:

```
focus <- as.POSIXct(c("2008-01-01",
    "2009-12-31"), tz = "UTC")
as <- subset(a, focus[1] <= time &
    time <= focus[2])
plot(as.section(as), which = "temperature",
    xtype = "time", ztype = "image")
```



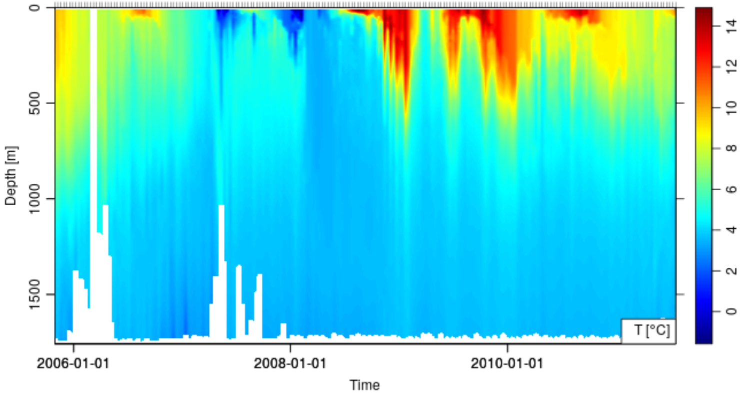
Subsetting data: by depth

```
as <- subset(a, pressure <= 1000 &
    pressure >= 200)
plot(as.section(as), which = "temperature",
    xtype = "time", ztype = "image")
```



Subsetting data: by dataMode

```
as <- subset(a, dataMode == "D")
plot(as.section(as), which = "temperature",
     xtype = "time", ztype = "image")
```



The future of oce and Argo

Potential additions will/could include

- splitting the argo-related code into a separate (but connected) package
- streamlined data discovery and download (within R)
- improved generic plot methods
- data products:
  - e.g. time series derived from float data
  - spatial objective mapping of fields
  - climatology creation (e.g. for limited regions or within a specified time range)
- improved BGC-related functionality (plotting, analysis, etc)
- data QA/QC tools

Manuscript in preparation (possibly for Frontiers in Marine Science)

Summary

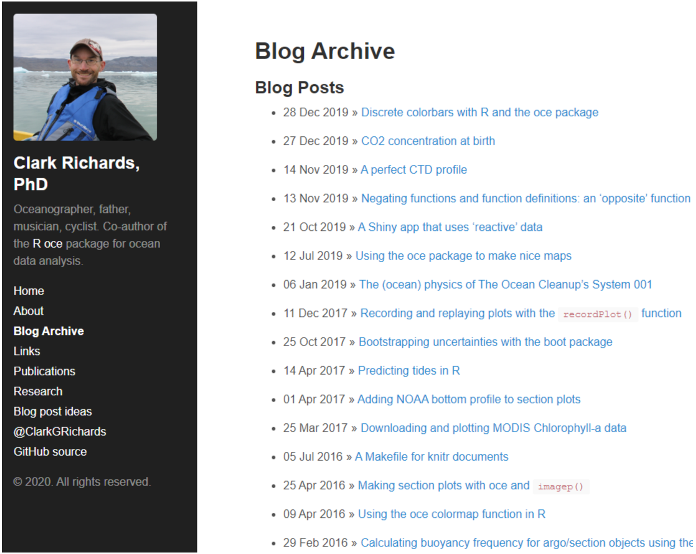
- R (and oce) provides a good ecosystem for working with oceanographic data (including Argo) due to its foundation in object orientation, class-specific methods, consistent data structures, and existing specialized oceanographic approaches

- Functionality already exists within oce for reading/working with Argo data, but lots of room to expand functionality

- R-based workflows can take advantage of other programming languages too: C/C++/Fortran can be built into packages directly, but also packages such as `reticulate` allow for integration of R and python objects and libraries (don't need to choose one or the other!)

Thanks!

`clark.richards@dfo-mpo.gc.ca` or `clarkrichards.org`



Clark Richards, PhD

Oceanographer, father, musician, cyclist. Co-author of the R oce package for ocean data analysis.

Home
About
Blog Archive
Links
Publications
Research
Blog post ideas
@ClarkGRichards
GitHub source

© 2020. All rights reserved.

Blog Archive

Blog Posts

- 28 Dec 2019 » Discrete colorbars with R and the oce package
- 27 Dec 2019 » CO2 concentration at birth
- 14 Nov 2019 » A perfect CTD profile
- 13 Nov 2019 » Negating functions and function definitions: an 'opposite' function
- 21 Oct 2019 » A Shiny app that uses 'reactive' data
- 12 Jul 2019 » Using the oce package to make nice maps
- 06 Jan 2019 » The (ocean) physics of The Ocean Cleanup's System 001
- 11 Dec 2017 » Recording and replaying plots with the `recordPlot()` function
- 25 Oct 2017 » Bootstrapping uncertainties with the boot package
- 14 Apr 2017 » Predicting tides in R
- 01 Apr 2017 » Adding NOAA bottom profile to section plots
- 25 Mar 2017 » Downloading and plotting MODIS Chlorophyll-a data
- 05 Jul 2016 » A Makefile for knitr documents
- 25 Apr 2016 » Making section plots with oce and `imagep()`
- 09 Apr 2016 » Using the oce colormap function in R
- 29 Feb 2016 » Calculating buoyancy frequency for argo/section objects using the