R code to generate chlorophyll-a concentration estimates with the eof method

Julien Laliberté

July 2020

This is a document explaining how the model developped in Laliberté et al. (2018) is built.

Inputs

You need two inputs,

- training set
- testing set

The output is the predicted chlorophyll

library

A training set is used to develop the model. It is simply a set of Rrs and associated Chl values, for example :

```
training_example1 <-
structure(list(id = c(8141L, 8489L, 11385L, 5627L, 9885L, 5027L,
7158L, 5027L, 5028L, 10812L), chl = c(0.26, 1.94, 4.33, 0.43,
1.8975, 0.635, 2.3025, 0.51, 0.6025, 0.245), Rrs_412 = c(0.00186400086658978,
0.00182400086669077, 0.000450000870159784, 0.00164600086714017,
0.00350800086243908, 0.00242000086518601, 0.00187000086657463,
0.000630000869705327, 0.000138000870947508, 0.00433400086035363
), Rrs_443 = c(0.00233000086541324, 0.00199200086626661, 0.000990000868796415,
0.00165800086710988, 0.00382400086164125, 0.00274000086437809,
0.00201400086621106, 0.00144000086766027, 0.00072200086947305,
0.00408600086097977), Rrs_490 = c(0.00282600086416096, 0.00196600086633225,
0.00161800086721087, 0.00161200086722602, 0.00352400086239868,
0.00311600086342878, 0.00220000086574146, 0.00204000086614542,
0.00124600086815008, 0.0038020008616968), Rrs_510 = c(0.00250600086496888,
0.00205400086611007, 0.00169800086700889, 0.00153000086743305,
0.00315600086332779, 0.00316000086331769, 0.00225200086561017,
0.00205600086610502, 0.00140200086775621, 0.00333600086287333
), Rrs_555 = c(0.00185600086660997, 0.00189000086652413, 0.00181600086671097,
0.00126400086810463, 0.00214000086589294, 0.00281800086418116,
0.0023640008653274, 0.00177200086682205, 0.0013380008679178,
0.0027960008642367), Rrs_670 = c(0.000248000870669784, 0.000220000870740478,
0.000292000870558695, 6.40008711343398e-05, 0.000324000870477903,
0.000432000870205229, 0.000628000869710377, 0.000168000870871765,
0.000144000870932359, 0.000304000870528398), type = c("chla",
"chla", "chla", "chla", "chla", "chla", "chla", "chla", "chla",
```

```
"chla")), row.names = c(NA, -10L), class = c("tbl_df", "tbl",
"data.frame"))



training_example1
```

```
##       id    chl      Rrs_412      Rrs_443     Rrs_490     Rrs_510     Rrs_555
## 1   8141 0.2600 0.0018640009 0.0023300009 0.002826001 0.002506001 0.001856001
## 2   8489 1.9400 0.0018240009 0.0019920009 0.001966001 0.002054001 0.001890001
## 3  11385 4.3300 0.0004500009 0.0009900009 0.001618001 0.001698001 0.001816001
## 4   5627 0.4300 0.0016460009 0.0016580009 0.001612001 0.001530001 0.001264001
## 5   9885 1.8975 0.0035080009 0.0038240009 0.003524001 0.003156001 0.002140001
## 6   5027 0.6350 0.0024200009 0.0027400009 0.003116001 0.003160001 0.002818001
## 7   7158 2.3025 0.0018700009 0.0020140009 0.002200001 0.002252001 0.002364001
## 8   5027 0.5100 0.0006300009 0.0014400009 0.002040001 0.002056001 0.001772001
## 9   5028 0.6025 0.0001380009 0.0007220009 0.001246001 0.001402001 0.001338001
## 10 10812 0.2450 0.0043340009 0.0040860009 0.003802001 0.003336001 0.002796001
##         Rrs_670 type
## 1  2.480009e-04 chla
## 2  2.200009e-04 chla
## 3  2.920009e-04 chla
## 4  6.400087e-05 chla
## 5  3.240009e-04 chla
## 6  4.320009e-04 chla
## 7  6.280009e-04 chla
## 8  1.680009e-04 chla
## 9  1.440009e-04 chla
## 10 3.040009e-04 chla
```

Here, ten matchups were retrieved for SeaWIFs, but you can replace this data frame with your own values. Note that we use `id` instead of lat/lon.

testing set

The testing set is just a set of Rrs values,

```
testing_example1 <-
structure(list(id = c(14896L, 10979L, 5401L, 14476L, 13290L,
8889L, 9285L, 14308L, 12961L, 4330L), date = structure(c(14737,
12953, 12574, 11779, 11123, 14840, 12482, 11569, 14317, 10853
), class = "Date"), Rrs_412 = c(0.000932000868942851,
0.00205000086612017, 0.000216000870750577, 0.00226600086557482,
0.00264000086463057, 0.000746000869412455, 0.00179000086677661,
0.0042260008606263, 0.00189200086651908, 0.000468000870114338
), Rrs_443 = c(0.00167600086706443, 0.00230000086548898, 0.000754000869392257,
0.00247000086505977, 0.00274000086437809, 0.00201800086620096,
0.00221400086570611, 0.00489600085893471, 0.00260200086472651,
0.000840000869175128), Rrs_490 = c(0.0023600008653375, 0.00269000086450433,
0.00139000086778651, 0.00268000086452957, 0.00283600086413571,
0.00316000086331769, 0.00282200086417106, 0.00516200085826313,
0.0027920008642468, 0.00149200086752899), Rrs_510 = c(0.00213200086591314,
0.00270000086447908, 0.00157000086733206, 0.00254800086486284,
0.00272200086442353, 0.00337800086276729, 0.00267400086454472,
```

```
0.00416400086078283, 0.00249200086500423, 0.00160400086724621
), Rrs_555 = c(0.00159800086726136, 0.00292800086390343, 0.00159400086727146,
0.00232400086542839, 0.00223000086566572, 0.00331000086293898,
0.00207400086605958, 0.00264000086463057, 0.00162600086719067,
0.0013260008679481), Rrs_670 = c(0.000132000870962656, 0.000762000869372059,
0.000186000870826319, 0.000774000869341762, 0.00022800087072028,
0.000566000869866912, 0.000358000870392061, 0.000338000870442556,
0.000360000870387012, 6.40008711343398e-05), type = c("rrs", "rrs", "rrs", "rrs", "rrs",
"rrs", "rrs", "rrs", "rrs", "rrs")), row.names = c(3694404L,
2549889L, 2221565L, 1485099L, 856018L, 3812659L, 2122325L, 1343338L,
3538414L, 640775L), class = "data.frame")



testing_example1
```

```
##            id       date      Rrs_412      Rrs_443     Rrs_490     Rrs_510
## 3694404 14896 2010-05-08 0.0009320009 0.0016760009 0.002360001 0.002132001
## 2549889 10979 2005-06-19 0.0020500009 0.0023000009 0.002690001 0.002700001
## 2221565  5401 2004-06-05 0.0002160009 0.0007540009 0.001390001 0.001570001
## 1485099 14476 2002-04-02 0.0022660009 0.0024700009 0.002680001 0.002548001
## 856018  13290 2000-06-15 0.0026400009 0.0027400009 0.002836001 0.002722001
## 3812659  8889 2010-08-19 0.0007460009 0.0020180009 0.003160001 0.003378001
## 2122325  9285 2004-03-05 0.0017900009 0.0022140009 0.002822001 0.002674001
## 1343338 14308 2001-09-04 0.0042260009 0.0048960009 0.005162001 0.004164001
## 3538414 12961 2009-03-14 0.0018920009 0.0026020009 0.002792001 0.002492001
## 640775   4330 1999-09-19 0.0004680009 0.0008400009 0.001492001 0.001604001
##             Rrs_555      Rrs_670 type
## 3694404 0.001598001 1.320009e-04  rrs
## 2549889 0.002928001 7.620009e-04  rrs
## 2221565 0.001594001 1.860009e-04  rrs
## 1485099 0.002324001 7.740009e-04  rrs
## 856018  0.002230001 2.280009e-04  rrs
## 3812659 0.003310001 5.660009e-04  rrs
## 2122325 0.002074001 3.580009e-04  rrs
## 1343338 0.002640001 3.380009e-04  rrs
## 3538414 0.001626001 3.600009e-04  rrs
## 640775  0.001326001 6.400087e-05  rrs
```

Ten values were randomly selected from SeaWIFs, but you can replace this data frame with your own values.

Function

The function in which the training set and testing set are used is named pred

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------------
```

```
## v ggplot2 3.3.0     v purrr   0.3.3
## v tibble  3.0.1     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## Warning: package 'tibble' was built under R version 3.6.2
```

```
## -- Conflicts -------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(psych)
```

```
##
## Attaching package: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
predict_chl <- function (testing_set, training_set) {

        all <- bind_rows(testing_set, training_set)
        #rm(testing_set)

        i <- which(all$type == "chla")
        n <- grepl("Rrs_[0-9]{3}$", names(all))
        my_pca <- princomp(log10(all[, n]))

        sc <- data.frame(my_pca$scores)

        df_chl <- data.frame(chl = all$chl[i], sc = sc[i, ])
        df_rrs <- data.frame(sc = sc[-i, ])

        ## Model with all PC
        my_lm <- lm(log10(chl) ~ ., data = df_chl)
        summary(my_lm)

        ## Select subset of PC
        step.lm <- step(my_lm, direction = "both", trace = 0)
        summary(step.lm)

        ## Predict chla with selected PC
        pred <- 10^predict(step.lm, newdata = df_rrs)

        #hist(log10(pred))
        #Ajust Chl EOF limits
        pred[which(pred>100)] <- 100 # ~
        pred[which(pred<0.001)] <- 0.001 # ~

        pred <- all %>%
            filter(type == "rrs") %>%
            mutate(chl_eof = pred)

        #Clean temp var
```

```
        pred <- pred  %>% mutate(
            type = NULL,
            chl = NULL) %>% mutate_at(vars(chl_eof), list(~round(., 3)))


        pred <- pred %>% dplyr::select(-contains('testing_set'),-chl_eof, everything())

        return(pred)
    }
```

When you execute the above function, you can predict chlorophyll-a. In this case, we have

```
predict_chl(testing_example1,training_example1)
```

```
##       id       date      Rrs_412      Rrs_443     Rrs_490     Rrs_510
## 1  14896 2010-05-08 0.0009320009 0.0016760009 0.002360001 0.002132001
## 2  10979 2005-06-19 0.0020500009 0.0023000009 0.002690001 0.002700001
## 3   5401 2004-06-05 0.0002160009 0.0007540009 0.001390001 0.001570001
## 4  14476 2002-04-02 0.0022660009 0.0024700009 0.002680001 0.002548001
## 5  13290 2000-06-15 0.0026400009 0.0027400009 0.002836001 0.002722001
## 6   8889 2010-08-19 0.0007460009 0.0020180009 0.003160001 0.003378001
## 7   9285 2004-03-05 0.0017900009 0.0022140009 0.002822001 0.002674001
## 8  14308 2001-09-04 0.0042260009 0.0048960009 0.005162001 0.004164001
## 9  12961 2009-03-14 0.0018920009 0.0026020009 0.002792001 0.002492001
## 10  4330 1999-09-19 0.0004680009 0.0008400009 0.001492001 0.001604001
##        Rrs_555      Rrs_670 chl_eof
## 1  0.001598001 1.320009e-04   0.333
## 2  0.002928001 7.620009e-04   2.736
## 3  0.001594001 1.860009e-04   1.112
## 4  0.002324001 7.740009e-04   3.426
## 5  0.002230001 2.280009e-04   0.516
## 6  0.003310001 5.660009e-04   0.534
## 7  0.002074001 3.580009e-04   0.951
## 8  0.002640001 3.380009e-04   0.188
## 9  0.001626001 3.600009e-04   1.084
## 10 0.001326001 6.400087e-05   0.369
```

And we have our prediction.