Getting started with the coastMDT package

Karina Nielsen

Contents

Installation

From Github

The easiest way to install the package is directly from Github by using the R devtools package.

1. Open R
2. Install the devtools library by typing

```
install.packages('devtools')
```

3. Install the coastMDT package

```
devtools::install_github("cavios/coastMDT/coastMDT")
```

From source

The instructions below describes how the package is installed from source

1. Download the source code (tar.gz) of the newest release from the Github page https://github.com/cavios/coastMDT under releases
2. Install the R packages that the package coastMDT depend on (see below under dependencies)

Linux

3. Open a shell in the directory containing the downloaded tar.gz file and write the following command. Replace "versionNo" with the actual version number.

```
R CMD INSTALL coastMDT_versionNo.tar.gz
```

Windows

3. Install Rtools https://cran.r-project.org/bin/windows/Rtools/
4. Use the R function `install.packages()` to install the package

```r
install.packages("coastMDT_versionNo.tar.gz", type="source", repos=NULL)
```

Or

3. In a CMD window (not in R), write

```
R CMD INSTALL coastMDT_versionNo.tar.gz
```

Dependencies

The packages depends on the following R libraries

- ncdf4
- smoothie
- fields
- FNN
- raster
- sp
- akima

Getting started with R

The section gives a short introduction to R, which is useful to new R users.

R tutorials can be found at the r-project web side https://cran.r-project.org/manuals.html

Help pages can be accessed by typing "?" in front of a given function. If we want to access the help for the function `plot` we write

```r
?plot
```

To exit R write

```r
q()
```

Introduction

coastMDT is an R package built to improve the estimate of the coastal mean dynamic topography (MDT). The package combine altimetry and tide gauge based MDT values. The tide gauge based MDT values are used to constrain the MDT values at the coastline and on land. An iterative average filter is used to smooth the raw MDT. Under each iteration the land values are reset. In this introduction, examples are given to illustrate how to use the package.

To load the package simply write:

```r
library(coastMDT)
```

Using the package "coastMDT", an example

This section gives a step by step guide on estimating the coastal MDT.

Load or read data

The Package contains several data sets, which are listed below. All the gridded data sets in the package are given on an 1/8 degree grid. The first cell is longitude 0 to 1/8 degree, latitude -90 to -90+1/8 degree and the order is east to west, then south to north. The MDT values at the at the tide gauges are given as a data frame inluding the columns; PSMSL station number, latitude, longitude, and MDT.

- **DTU15MSS**: DTU15 mean sea surface on 1/8 degree grid (0-360 degree).
- **eigen6c4r**:Geoid model based on EIGEN-6C4 on 1/8 degree grid (0-360 degree).
- **landmask8**:Land/ocean mask on 1/8 degree grid (0-360 degree).
- **difmss15eig6c4r**: Raw MDT based on the mean sea surface DTU15MSS and the geoid model eigen6c4r.
- **dDTU15MSS_ref2003_2007**: Grid to transform the DTU15 MSS (DTU15MSS) to the MSS of the reference period 2003-2007. The grid is defined on 1/8 degree grid (0-360 degree).
- **ibCor5Y_2003_2007**: Inverse barometer corrections for the 5-year reference period 2003-2007 on 1/8 degree grid (0-360 degree).
- **dacCor5Y_2003_2007**: Dynamic atmosphere correction for the 5-year reference period 2003-2007 on 1/8 degree grid (0-360 degree).
- **mean2TF_AddThis**: Grid to go from the mean tide system to the tide free system. The grid is defined on 1/8 degree grid (0-360 degree).
- **TF2mean_AddThis**: Grid to go from the tide free system to the mean tide system. The grid is defined on 1/8 degree grid (0-360 degree).
- **MDT_eigen6c4r_2003_2007**: MDT values based on tide gauge data.

The build in data sets can easely be loaded into R as demonstrated in the example below

```r
data(DTU15MSS)
DTU15MSS[200:203, 100:103]
```

```
##        [,1]   [,2]   [,3]   [,4]
## [1,] 12.391 12.741 13.118 13.550
## [2,] 12.460 12.812 13.188 13.617
## [3,] 12.525 12.882 13.257 13.682
## [4,] 12.588 12.948 13.322 13.744
```

The users can also import their own data in which case other appropriate R functions should be used depending on the format of the data. The package has a simple function `readncdf1var` to read one-variable NetCDF files.

Data for constructing the MDT

The data needed to construct the coastal MDT are listed below

```r
data(DTU15MSS) # MSS
data(eigen6c4r)# Geoid model
data(landmask8)# land/ocean mask
data(dDTU15MSS_ref2003_2007) # grid to correct the MSS to the reference period.
data(ibCor5Y_2003_2007)# Inverse barometer correction
```

```r
data(MDT_eigen6c4r_2003_2007)# Tide gauge MDT
TG<-MDT_eigen6c4r_2003_2007
```

Constructing the raw MDT

The geodetic MDT ξ is constructed from the MSS and the geoid.

$$\xi = MSS - N, \tag{1}$$

where N is the geoid. In R the two grids are simply subtracted.

```r
data(DTU15MSS)
data(eigen6c4r)
MDTraw<-DTU15MSS-eigen6c4r
```

When working with different data types such as altimetry and tide gauge data it is important to make sure that the data are in the same height reference system, time period and that the same corrections has been applied. In this example we will work with tide gauge data in covering the period 2003-2007. Hence the DTU15 MSS needs to be converted to the same reference period.

The inverse barometer correction is applied to the MSS. Hence it must be re-added to the MSS in case the tide gauge data are not IB corrected. In the example below the MSS is corrected for the reference period and the IB effect is re-added to the MSS.

```r
MDTraw<-DTU15MSS+dDTU15MSS_ref2003_2007+ibCor5Y_2003_2007-eigen6c4r
```

The 5 year (2003-2007) IB correction and the period correction to the MSS for the east coast of the US is shown on figure 1

The altimetry MSS is referenced relative to the TOPEX ellipsoid, while the tide gauge data is referenced relative to WGS84. Hence we convert the tide gauge data to the TOPEX reference ellipsoid. The following expression approximates the height difference Δh assuming that the difference in latitude is very small.

$$\Delta h = -((aTop - aWGS)cos(\phi)^2 + (bTop - bWGS)sin(\phi)^2). \tag{2}$$

Here ϕ is the latitude, $aTop$ and $aWGS$ is the equatorial radius of TOPEX and WGS84, respectively, and $bTop$ and $bWGS$ is the polar radius of TOPEX and WGS84, respectively.

The above expression has been implemented in the function `wgs2topCorr`. Hence the tide gauge data is converted to TOPEX ellipsoid height with the following command

```r
TG[,4]<-TG[,4]+wgs2topCorr(TG[,2])
```

The two data sets should now be given in the same height reference system, in the same reference period, and the same correction should be applied.

Extraction a subsection

The available data sets are global,but the user has the possibility to extract regional grids with the function `getSubGrid`. In the example below the East coast of the US is extracted from the raw MDT and the land/ocean mask. The tide gauge data can be extracted with the function `getSubTG`.

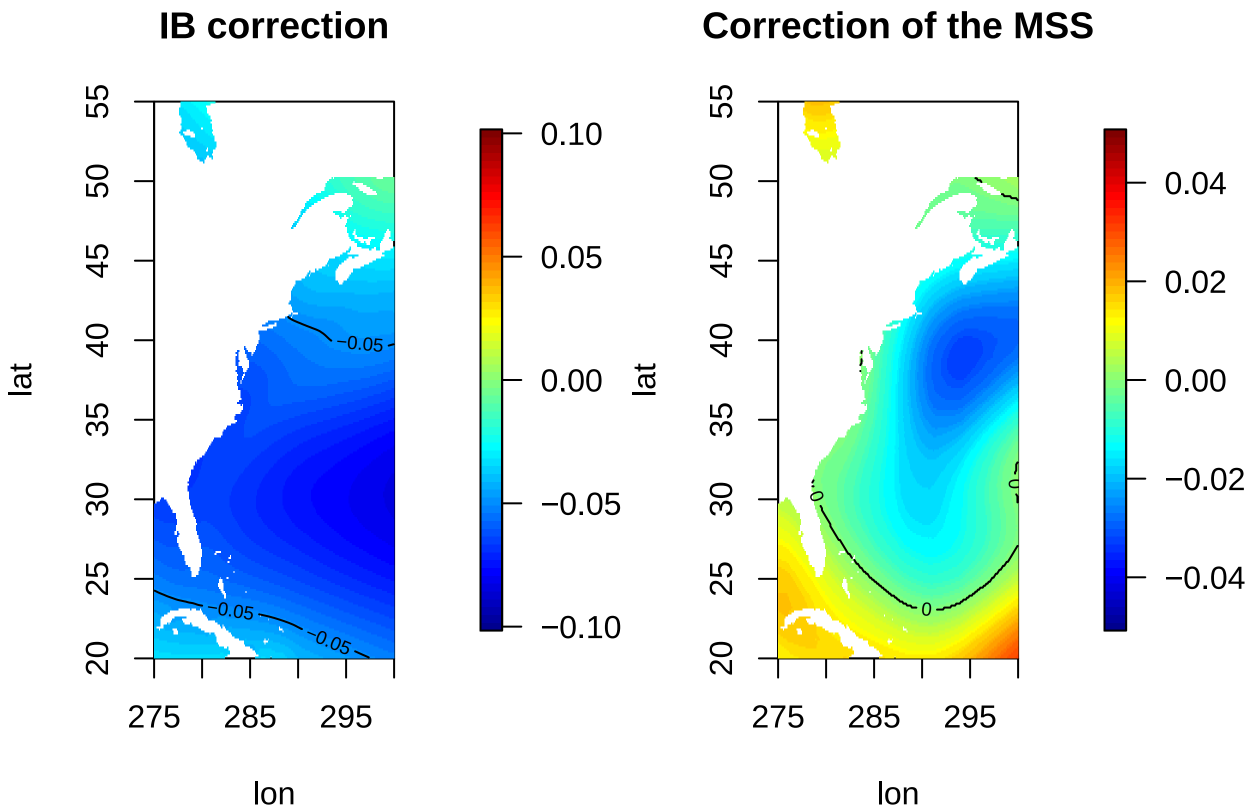

Figure 1: Left: 5 year (2003-2007) IB correction. Right: 5 year (2003-2007) correction to the MSS

```
#Region of interest; here the East coast of the US
lonlim<-c(275,300)
latlim<-c(20,55)

#sub grids and data
rawUS<-getSubGrid(MDTraw,lonlim,latlim)
mask<-getSubGrid(landmask8,lonlim,latlim)
TGsub<-getSubTG(TG,lonlim,latlim)
```

Defining the land values

MDT Land values are estimated with the function `getLandVal`. The land values may be based on tide gauge MDT values, the altimetry alone, or a combination of both. This option is specified with the parameter `type=("tg","alt", or "both")`. In the example below both tige gauge values and altimetry is used to contrain the land MDT values, see figure 2.

```
mylandTG_ALT <- getLandVal(rawUS$g, mask$g, lonlim, latlim, TG = TGsub, type = "both",
    intMethod = "lin")
```

```
plotMDT(mylandTG_ALT, c(-0.5, 1), lonlim, latlim, addContour = FALSE)
```

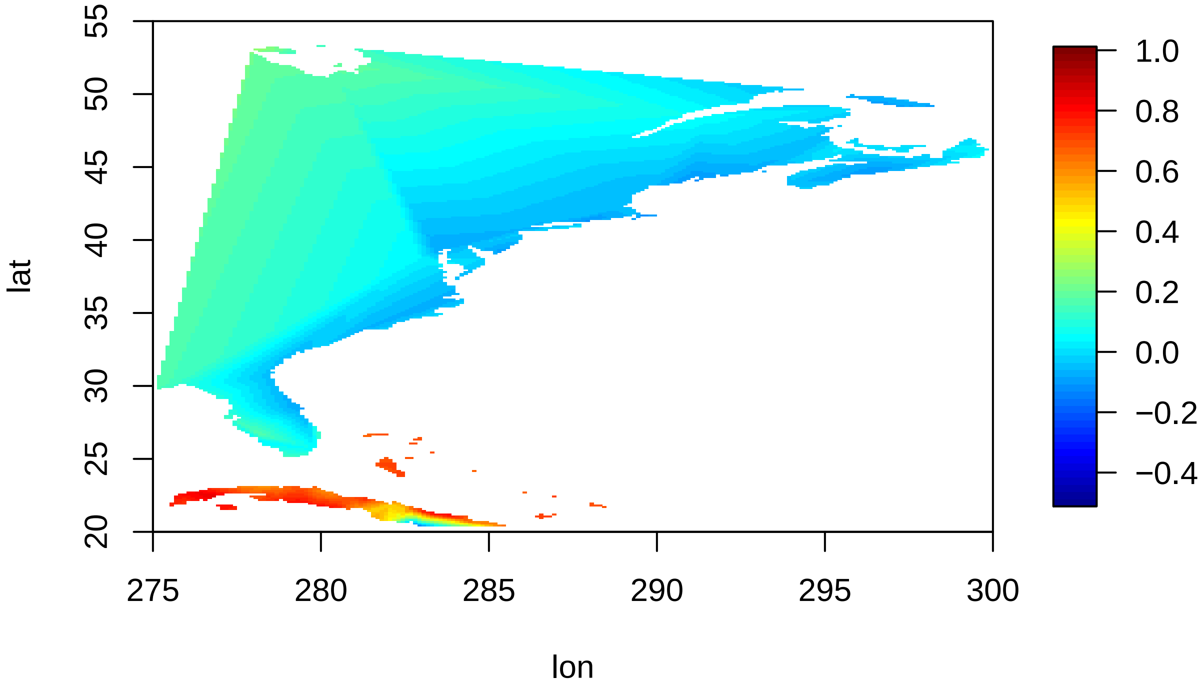

Figure 2: Land values based on tide gauge and altimetry data

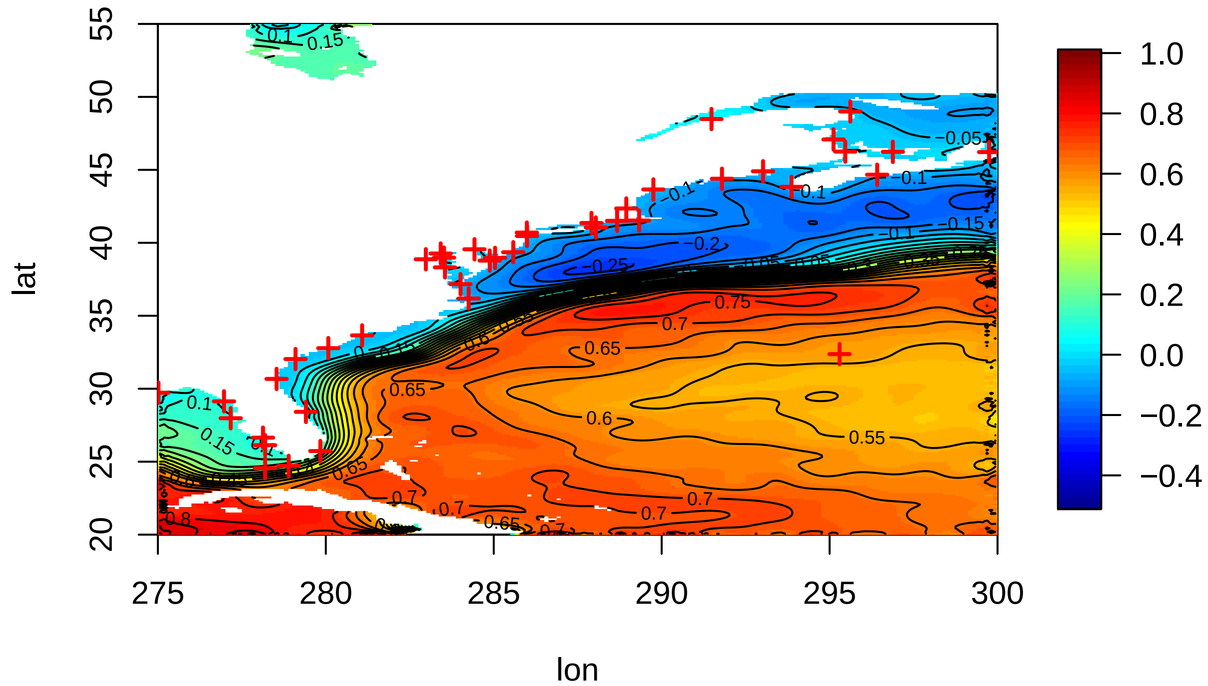

Figure 3: The filtered MDT. The red crosses displays the positions of the tide gauges

Filtering

After the land values are estimated we can start to filter the raw MDT. The package applies an iterative average filter. After each iteration the land values are reset to the original values. It is possible to vary the radius of the filter and the number of iterations that is used. In the example below the default values are used.

```
boxTG_ALT <- iterativeAveSmoother(rawUS$g, mask$g, t(mylandTG_ALT), latlim)
```

Plotting

The package includes a simple plotting function, so the output MDT can easily be displayed. The example below displays the MDT for the recurring example

```
plotMDT(t(boxTG_ALT), c(-0.5, 1), lonlim, latlim, conlev = 0.05)
points(TGsub[, 3:2], col = "red", pch = 3, lwd = 2)
```

Summary

The compleete example is summarized below

```r
library(coastMDT)
# read data
data(landmask8)  #land mask
# the raw MDT
data(DTU15MSS)
data(eigen6c4r)
MDTraw <- DTU15MSS + dDTU15MSS_ref2003_2007 + ibCor5Y_2003_2007 - eigen6c4r
data(MDT_eigen6c4r_2003_2007)
TG <- MDT_eigen6c4r_2003_2007
TG[, 4] <- TG[, 4] + wgs2topCorr(TG[, 2])  # WGS84 -> TOPEX ellipsoid

# region of interest
lonlim <- c(275, 300)
latlim <- c(20, 55)

# sub grids and data
mask <- getSubGrid(landmask8, lonlim, latlim)
rawUS <- getSubGrid(MDTraw, lonlim, latlim)
TGsub <- getSubTG(TG, lonlim, latlim)

# Estimation of land data
mylandTG_ALT <- getLandVal(rawUS$g, mask$g, lonlim, latlim, TG = TGsub, type = "both",
    intMethod = "lin")

# Filtering
boxTG_ALT <- iterativeAveSmoother(rawUS$g, mask$g, t(mylandTG_ALT), latlim)

# plotting
plotMDT(t(boxTG_ALT), c(-0.5, 1), lonlim, latlim, conlev = 0.05)
```