# Package: tsHydro (via r-universe)

October 13, 2026

**Title** Reconstruct water level time series from satellite altimetry data

**Version** 0.0.0.9000

**Description** What the package does (one paragraph).

**License** `use_mit_license()`, `use_gpl3_license()` or friends to pick a license

**Imports** TMB

**LinkingTo** TMB, RcppEigen

**LazyData** TRUE

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.1.1

**Repository** https://cavios.r-universe.dev

**Date/Publication** 2024-09-03 22:06:46 UTC

**RemoteUrl** https://github.com/cavios/tshydro

**RemoteRef** HEAD

**RemoteSha** 110f81c430ad2eb89220448d28667742648be69c

**RemoteSubdir** tsHydro

## Contents


export.tsHydro . . . . . 2
get.TS . . . . . 2
plot.tsHydro . . . . . 4
summary.tsHydro . . . . . 5

**Index** **6**

`export.tsHydro` *Export output*

## Description

This function saves the predicted water levels to a file

## Usage

```
export.tsHydro(x, filename = "ts.dat", exportPar = FALSE)
```

## Arguments

| | |
|---|---|
| `x` | An object of class "tsHydro" |
| `filename` | Name of output file |
| `exportPar` | Logic variable to specify if the estimated model parameters are saved to a file "tsPar.dat". |
| `addError` | To add error bars |

## Value

The following elements

- oupfile("ts.dat") A text file that contains three colunms; "time", "wl", "wlsd". "time" is the time of each pass, where the water level is estimated. "wl" is the estimated water level and "wlsd" is the standard deviation of the estimated water level.
- "tspar.dat" A text file that contains the optimized model parameters

## Examples

```
data(lakelevels)
fit<-get.TS(lakelevels)
export.tsHydro(fit,file="myTS.dat",exportPar=TRUE)
```

`get.TS` *Reconstruct water level*

## Description

Estimate the model parameters and return the estimated water levels

## Usage

```
get.TS(
  dat,
  init.h = 0,
  init.logsigmarw = 0,
  init.logSigma = getSigmaInit(dat$satid),
  bias = rep(0, length(unique(dat$satid)) - 1),
  init.logit = log(0.1/(1 - 0.1)),
  priorHeight = numeric(0),
  priorSd = numeric(0),
  estP = FALSE,
  silent = TRUE,
  weights = rep(1, nrow(dat)),
  varPerTrack = FALSE,
  varPerQuality = FALSE,
  newdat = NULL,
  ...
)
```

## Arguments

| | |
|---|---|
| dat | A data.frame containing at least the columns: time, height, and track |
| init.h | Initial value for the mean water levels. The default value is 0. |
| init.logsigmarw | Initial value for the log of the standard deviation of the random walk |
| init.logSigma | Initial value for the log of the standard deviation of the random walk |
| bias | Optional, vector of length N-1 with Initial values of the bias estimates, where N is the numbers of satellite missions used. To estimate the bias "dat" must have a column "satid" with the ids of the satellites for each observation, 0,1,2,3,..,N-1. The number of bias estimates is N-1. The bias estimates is w.r.t. the satellite with the largest id. If dat$satid is provided the observation standard deviation is estimated pr satellite |
| init.logit | Initial value for the log of the standard deviation of the observation noise |
| priorHeight | ... |
| priorSd | ... |
| estP | A logic value FALSE if the outlier fraction is kept fixed at its initial value |
| weights | Optional vector of weights. |
| varPerTrack | Optional, a logic value: If TRUE, an observation standard deviation is "logSigma" is estimated per track. |
| varPerQuality | Optional, a logic value: If TRUE, an observation standard deviation is "logSigma" is estimated per quality id. If this option is used "dat" must have a column named "qf" |
| newdat | Optional, a data frame which at least should include a column named "time", containing the time in decimal years where the modeled water level is predicted. newdat may also include a column named "group" with a group id for each |

| | |
|---|---|
| | observation. Groups could be based on month, years, or something else. If "group" is provided a average water level pr group is also provided |
| exportPar | Logic variable to specify if the estimated model parameters are saved to a file "tsPar.dat". |

## Details

The function can handle the observation based standard deviation in different ways; either pr satellite, pr track, or pr quality. However, these options cannot be used together.

## Value

An object of class "tsHydro" with the following elements

- oupfile("ts.dat") A text file that contains three colunms; "time", "wl", "wlsd". "time" is the time of each pass, where the water level is estimated. "wl" is the estimated water level and "wlsd" is the standard deviation of the estimated water level.
- "tspar.dat" A text file that contains the optimized model parameters

## Examples

```
data(lakelevels)
fit<-get.TS(lakelevels)
```

---

# plot.tsHydro *A Plot function*

---

## Description

Plot an object returned by the function get.TS()

## Usage

```
## S3 method for class 'tsHydro'
plot(
  x,
  addRawDat = TRUE,
  addLine = TRUE,
  addError = FALSE,
  zoomOut = FALSE,
  lwd = 4,
  col = "blue",
  ...
)
```

### Arguments

| | |
|---|---|
| x | Object returned by get.TS() |
| addLine | To add a line between the points that represents the estimated water levels. |
| addError | To add error bars |
| zoomOut | To zoom out. This option creates a plot which displays the range of the water level data. |
| lwd | Line width |
| col | Color |
| ... | Additional argumants to plot |
| addRawdat | Adds the data, which the estimated water levels are based on |

### Examples

```
data(lakelevels)
fit<-get.TS(lakelevels)
#Plot with error bars
plot(fit,addError=TRUE,col='blue')
#plot that includes the water level data
#and displays the entire data range
plot(fit,zoomOut=TRUE,col='red')
```

---

## summary.tsHydro *Summary of output*

---

### Description

This function presents a summary of the output

### Usage

```
## S3 method for class 'tsHydro'
summary(x)
```

### Arguments

| | |
|---|---|
| x | An object of class "tsHydro" |

### Value

Summary of output

# Index

∗ **plot**
    export.tsHydro, 2
    plot.tsHydro, 4
∗ **series**
    get.TS, 2
∗ **time**
    get.TS, 2

export.tsHydro, 2

get.TS, 2

plot.tsHydro, 4

summary.tsHydro, 5