u* threshold dectection

Thomas Wutzler

Loading [MathJax]/jax/output/HTML-CSS/jax.js

Importing the half-hourly data

Reads a text file with data of the year 1998 from the Tharandt site.

```
library(REddyProc)
EddyData <- fLoadTXTIntoDataframe("DETha98.txt")
```

```
## Loaded file DETha98.txt with the following variables (units):
```

```
##  *** Year(-) DoY(-) Hour(-) NEE(umolm-2s-1) LE(Wm-2) H(Wm-2) Rg(Wm-2) Tair(degC) Tsoil(degC
```

```
## Number of '-9999' convertered to NA: 11691
```

Example Data

```
head(EddyData)
```

```
##   Year DoY Hour   NEE   LE      H Rg Tair Tsoil    rH VPD Ustar
## 1 1998   1  0.5 -1.21 1.49 -11.77  0  7.4  4.19 55.27 4.6  0.72
## 2 1998   1  1.0  1.72 3.80 -13.50  0  7.5  4.20 55.95 4.6  0.52
## 3 1998   1  1.5    NA 1.52 -18.30  0  7.1  4.22 57.75 4.3  0.22
## 4 1998   1  2.0    NA 3.94 -17.47  0  6.6  4.23 60.20 3.9  0.20
## 5 1998   1  2.5  2.55 8.30 -21.42  0  6.6  4.22 59.94 3.9  0.33
## 6 1998   1  3.0    NA 1.33 -20.55  0  6.5  4.21 59.25 4.0  0.15
```

Initializing the REddyProc class

```
# create R timestamp from columns Year, DoY and Hour
EddyDataWithPosix <- fConvertTimeToPosix(
  EddyData, 'YDH',Year = 'Year',Day = 'DoY', Hour = 'Hour')
```

```
## Converted time format 'YDH' to POSIX with column name 'DateTime'.
```

```
EProc <- sEddyProc$new(
  'DE-Tha', EddyDataWithPosix, c('NEE','Rg','Tair','VPD', 'Ustar'))
```

```
## New sEddyProc class for site 'DE-Tha'
```

```
EProc$sSetLocationInfo(LatDeg = 51.0, LongDeg = 13.6, TimeZoneHour = 1)
```

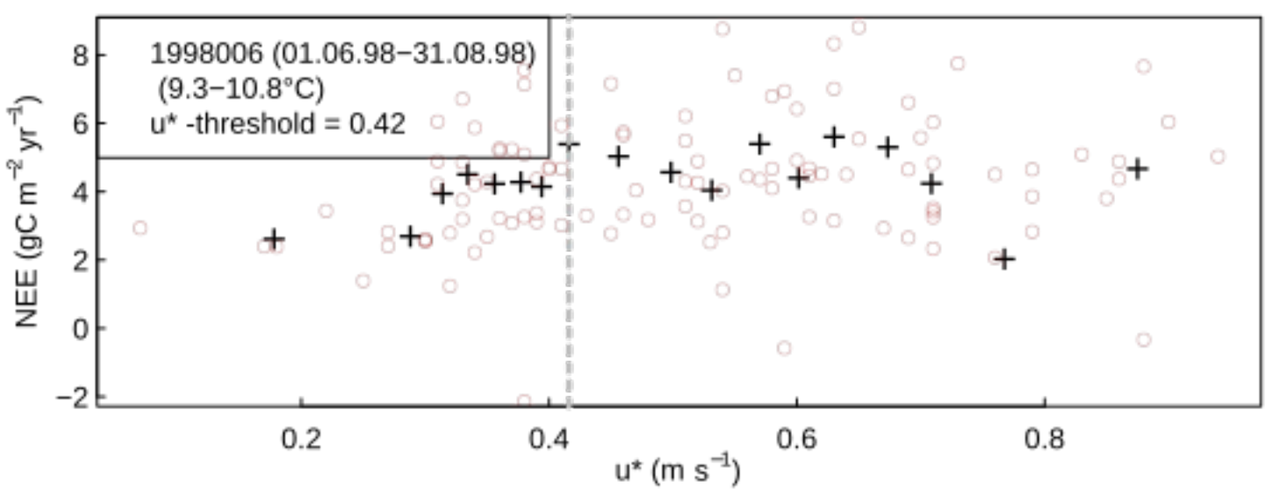
Getting help: "?"

```
# display overview help of REddyProc
?REddyProc

# display help of a specific function
?fConvertTimeToPosix

# display help of a method of the REddyProc class
# prepend by 'sEddyProc_'
?sEddyProc_sEstimateUstarScenarios
```

Problem: NEE biased low at low u*



At low frction velocity (u_*) the NEE is biased low. Threshold detection estimates the u_* above with there is no bias. Discarding periods with low uStar is one of the largest sources of uncertainty in aggregated fluxes.

Estimating the uStar threshold distribution

The estimate of the u_* theshold is uncertain. Hence, several quantiles of the distribution of the uncertain uStar threshold are estimated by a bootstrap.

The friction velocity, uStar, needs to be in column named "Ustar" of the input dataset.

```
EProc$sEstimateUstarScenarios(
  nSample = 20L, probs = c(0.05, 0.5, 0.95))
```

Inspect the estimated u_* threshold

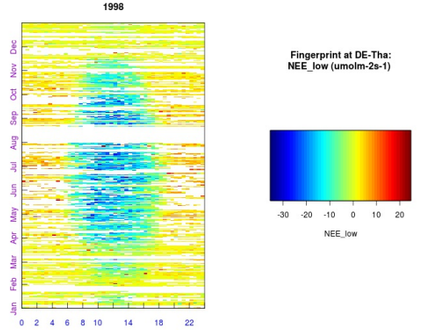
```
(uStarTh <- EProc$sGetEstimatedUstarThresholdDistribution())
```

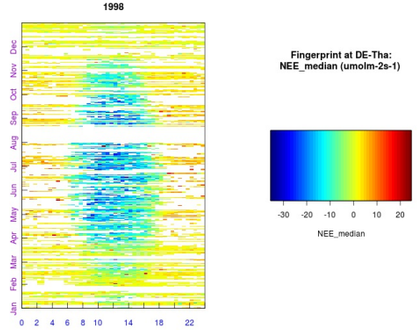
```
##   aggregationMode seasonYear  season     uStar        5%       50%
## 1          single         NA    <NA> 0.4162500 0.3963710 0.4481250
## 2            year       1998    <NA> 0.4162500 0.3963710 0.4481250
## 3          season       1998 1998001 0.4162500 0.3963710 0.4481250
## 4          season       1998 1998003 0.4162500 0.3481563 0.4059808
## 5          season       1998 1998006 0.3520000 0.3440556 0.3812196
## 6          season       1998 1998009 0.3369231 0.3134888 0.3875000
## 7          season       1998 1998012 0.1740000 0.2851500 0.4231250
##         95%
## 1 0.5776313
## 2 0.5776313
## 3 0.5776313
## 4 0.5687276
## 5 0.4475625
## 6 0.4888738
## 7 0.5753171
```

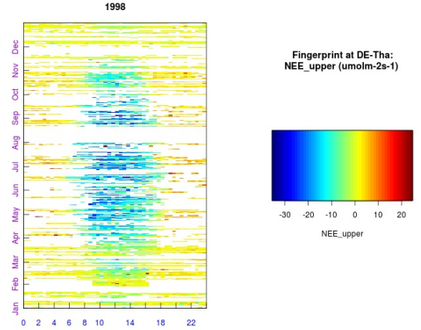
Effects on gap coverage

With increasing u_* threshold, the fraction of records that are marked as gaps increases. Hence, all the gappfilling, flux-partitioning, and aggregation need to by repeated for diffrent u_* threshold estimates.

low u_* threshold	medium u_* threshold	high u_* threshold







Aggregating threshold estimates

There threshold may change across time. Hence, there is an estimate for different time periods, i.e. seasons. To be conservative, by default gapfilling uses the annually aggregated values.

```
EProc$sGetUstarScenarios()
```

```
##    season   uStar      U05      U50      U95
## 1 1998001 0.41625 0.396371 0.448125 0.5776313
## 2 1998003 0.41625 0.396371 0.448125 0.5776313
## 3 1998006 0.41625 0.396371 0.448125 0.5776313
## 4 1998009 0.41625 0.396371 0.448125 0.5776313
## 5 1998012 0.41625 0.396371 0.448125 0.5776313
```

Using seasonal estimates

One can use seasonal estimates, instead of the annual aggregates.

EProc$sGetUstarScenarios()

```
EProc$useSeaonsalUStarThresholds()
EProc$sGetUstarScenarios()
```

```
##    season     uStar       U05       U50       U95
## 3 1998001 0.4162500 0.3963710 0.4481250 0.5776313
## 4 1998003 0.4162500 0.3481563 0.4059808 0.5687276
## 5 1998006 0.3520000 0.3440556 0.3812196 0.4475625
## 6 1998009 0.3369231 0.3134888 0.3875000 0.4888738
## 7 1998012 0.1740000 0.2851500 0.4231250 0.5753171
```

User-defined seasons

The default seasons are (Dec,Jan,Feb) (Mar,Apr,Mai) (Jun,Jul,Aug) and (Sep,Oct,Nov). If you know at which dates the surface roughness changes, e.g. with harvest you can tell the seasonal breakes.

```
seasonStarts <- data.frame( c(70,210,320), 1998 )
seasonFactor <- usCreateSeasonFactorYdayYear(
  EddyDataWithPosix$DateTime - 15*60, starts = seasonStarts)
#EProc$sEstimateUstarScenarios(
#   seasonFactor = seasonFactor
#   nSample = 100L, probs = c(0.05, 0.5, 0.95))
```