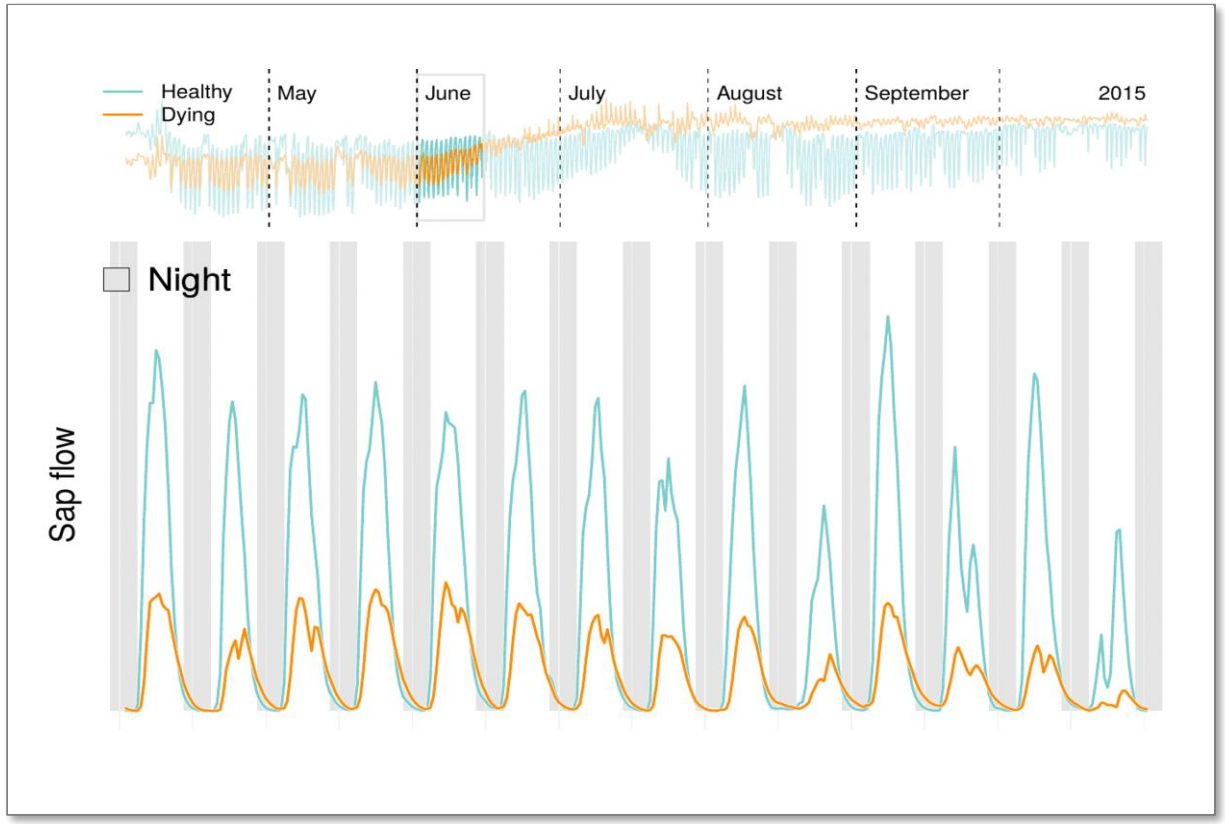
WK 37 - Advanced Tree Physiological Data Processing in R

Friday, August 19, 2022 10:00 AM – 11:30 AM EDT Location: 514C

ESACSEE

Goal?

Facilitate the work with raw time series data and make methods for their processing accessible and reproducible



Christoforos Pappas — Πανεπιστήμιο Πατρών, University of Patras

Richard L. Peters — University of Basel

Alexander G. Hurley — GFZ Helmholtz Centre Potsdam

Roman Zweifel — WSL

ESACSEE

Outline

*(1) Data **handling & cleaning***

`datacleanr`

https://the-hull.github.io/datacleanr

*(2) Sap flow data **processing***

`TREX`

https://the-hull.github.io/TREX

*(3) Dendrometer data **processing***

`treenetproc`

https://github.com/treenet/treenetproc

(4) Hands-on & Q/A

WK 37 - Advanced Tree Physiological Data Processing in R

ESACSEE

Outline

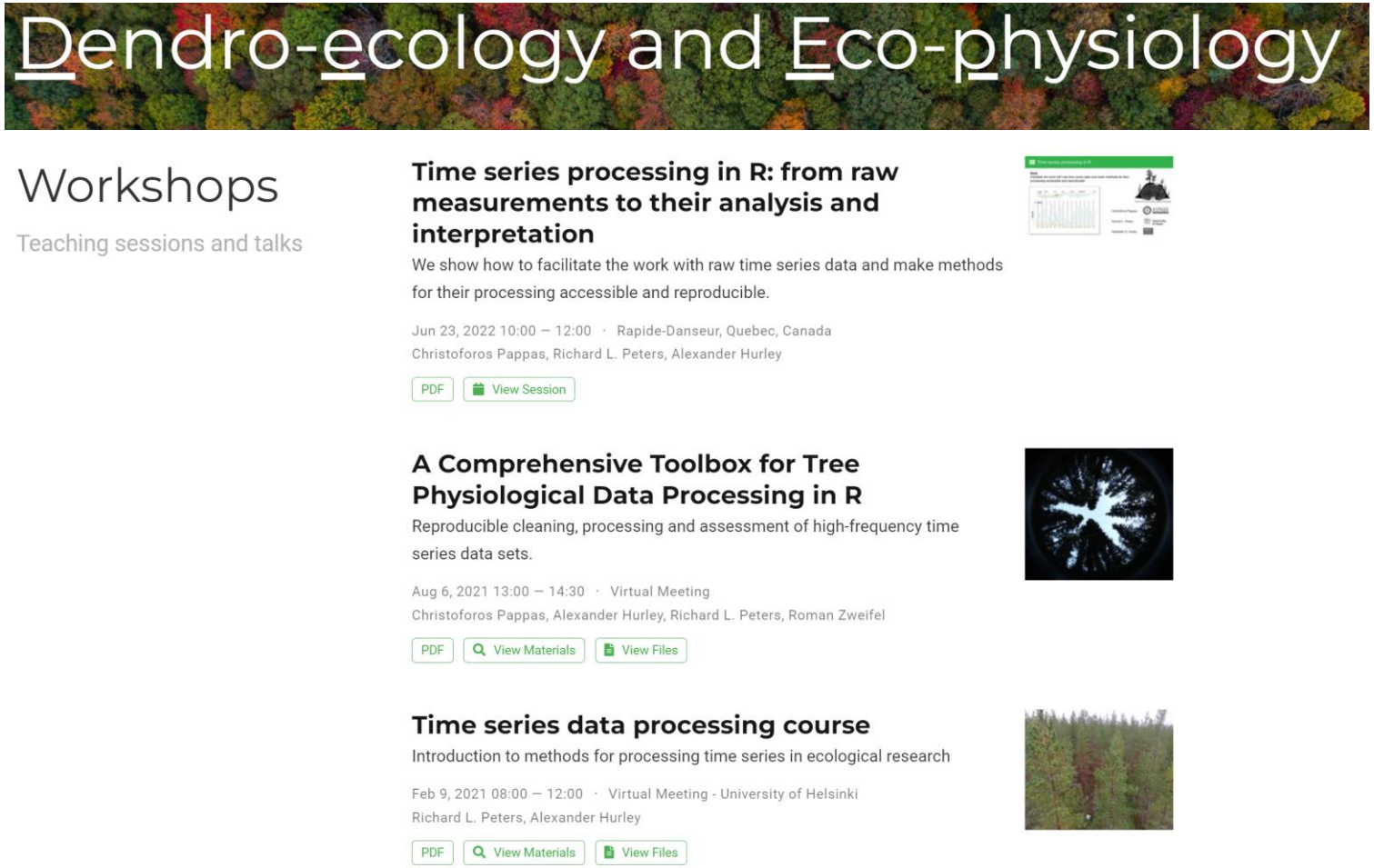
*(1) Data **handling & cleaning***

*(2) Sap flow data **processing***

*(3) Dendrometer data **processing***

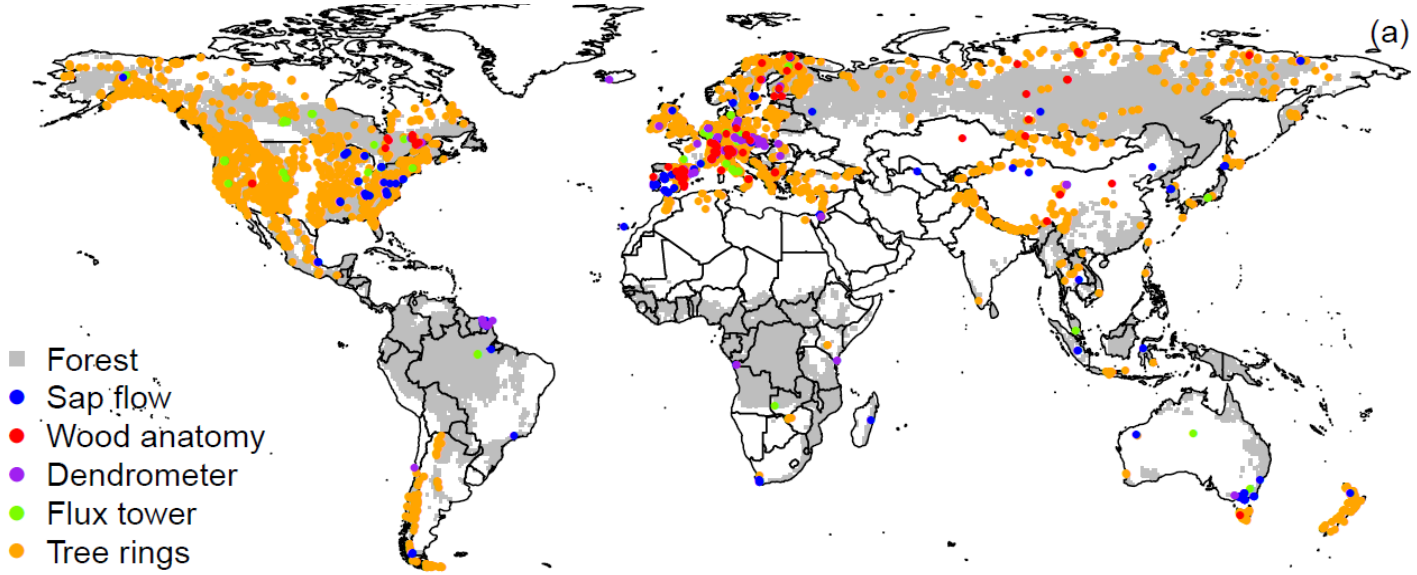
(4) Hands-on & Q/A

https://deep-tools.netlify.app/



Relevance

Ecological research is becoming increasingly data-rich!



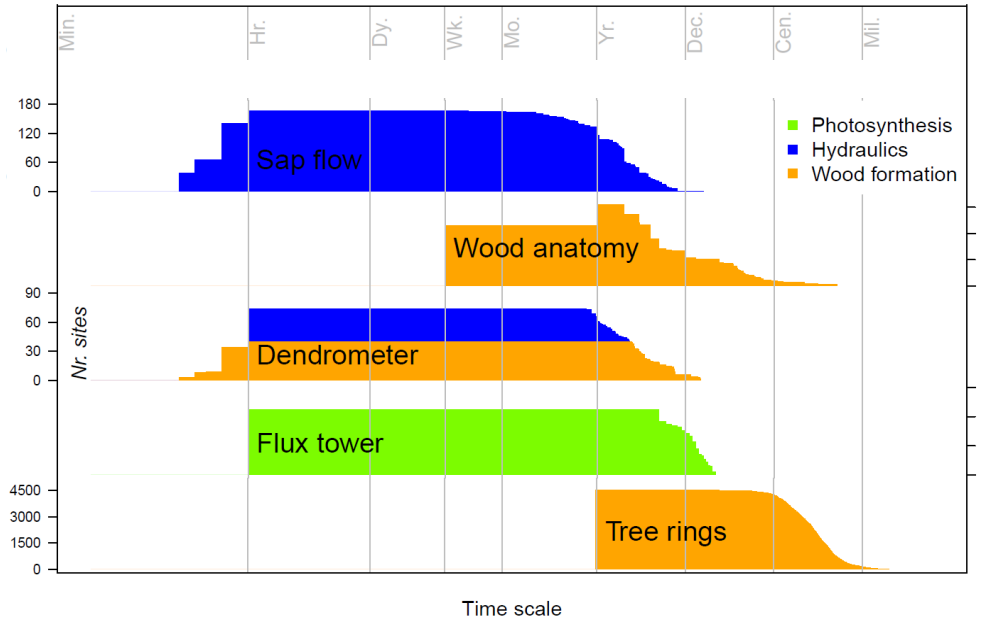


Fig. Global distribution of sites with relevant tree physiological measurements.
Source: Peters (2018) doi: https://doi.org/10.5451/unibas-007085812

Relevance

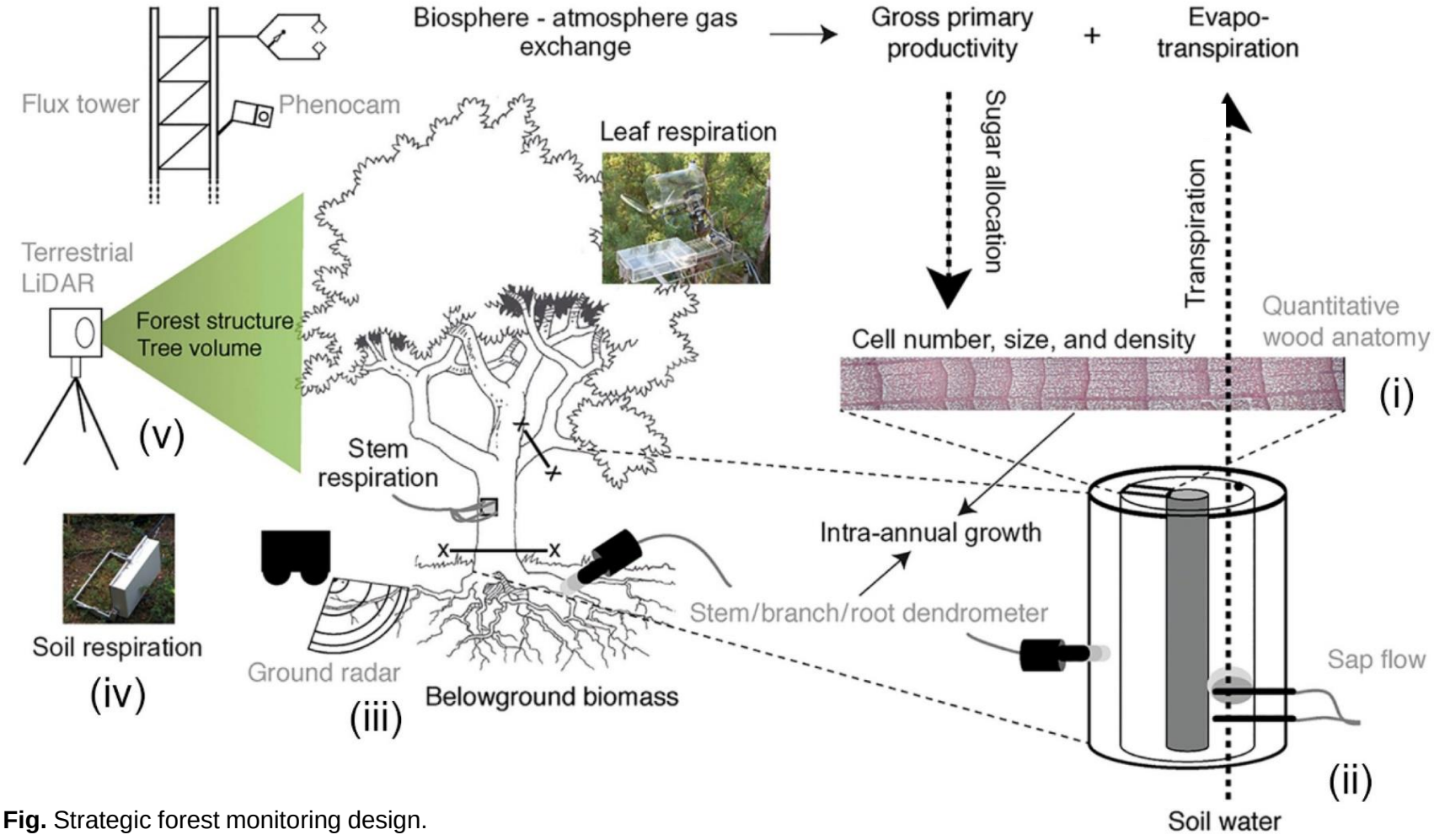
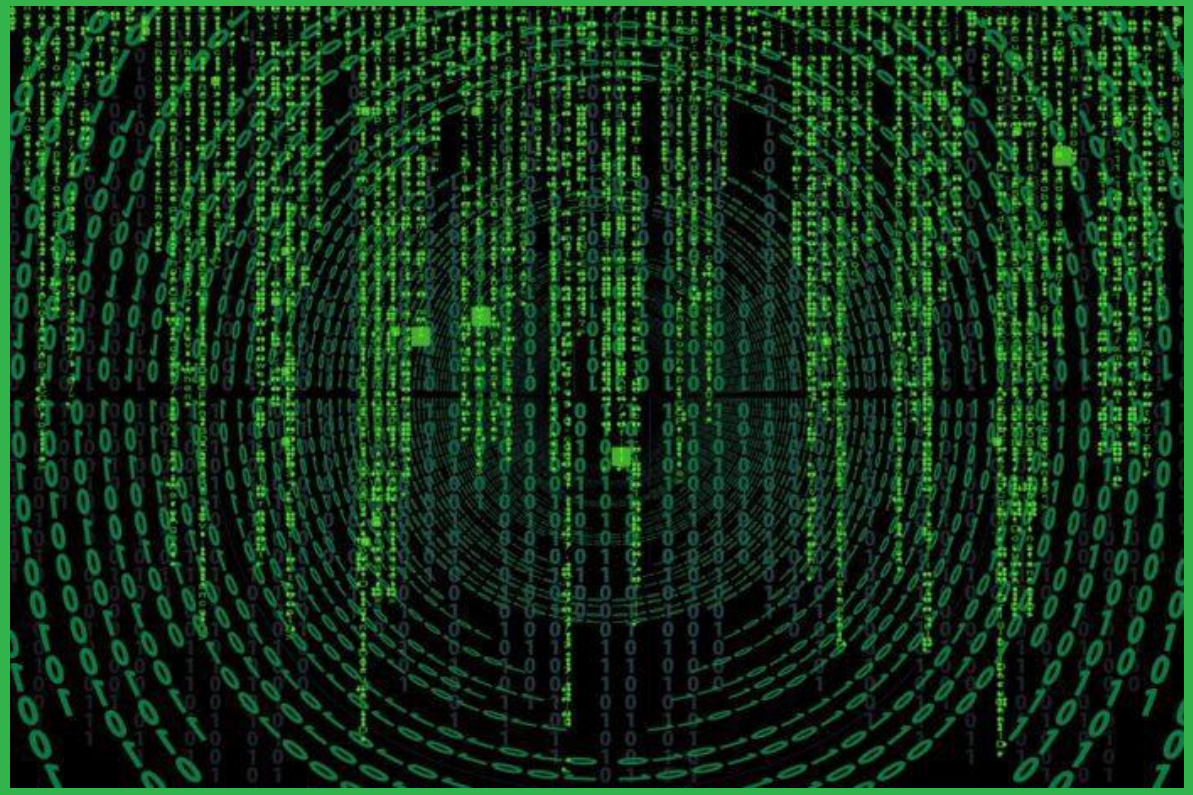


Fig. Strategic forest monitoring design.
Source: Babst et al. (2021) doi: https://doi.org/10.1016/j.tplants.2020.10.002

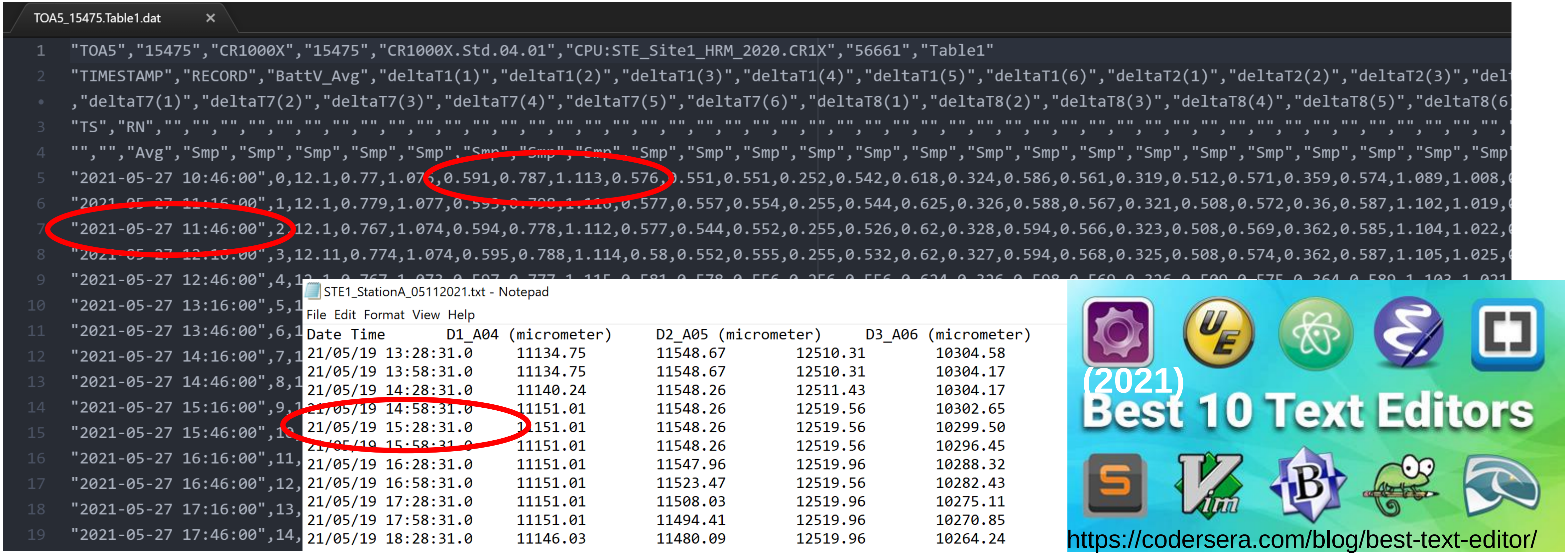
Data handling

Goal? *Raw data import in R, data organization, aggregation, format homogenization.*



Raw data

- Typically, text files/spreadsheets
- Delimited formats (tab, space, …), data types (numeric, strings, factors, …)
- Quick view with a text editor



Time series handling

Data import in R

- base R
- tidyverse

```
read.table(file, header = FALSE, sep = "", quote = "\"'",
           dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"),
           row.names, col.names, as.is = !stringsAsFactors,
           na.strings = "NA", colClasses = NA, nrows = -1,
           skip = 0, check.names = TRUE, fill = !blank.lines.skip,
           strip.white = FALSE, blank.lines.skip = TRUE,
           comment.char = "#",
           allowEscapes = FALSE, flush = FALSE,
           stringsAsFactors = default.stringsAsFactors(),
           fileEncoding = "", encoding = "unknown", text, skipNul = FALSE)

read.csv(file, header = TRUE, sep = ",", quote = "\"",
         dec = ".", fill = TRUE, comment.char = "", …)

read.csv2(file, header = TRUE, sep = ";", quote = "\"",
          dec = ",", fill = TRUE, comment.char = "", …)

read.delim(file, header = TRUE, sep = "\t", quote = "\"",
           dec = ".", fill = TRUE, comment.char = "", …)

read.delim2(file, header = TRUE, sep = "\t", quote = "\"",
            dec = ",", fill = TRUE, comment.char = "", …)
```

https://www.rdocumentation.org/packages/utils/versions/3.6.2/topics/read.table

readr

- `read_csv()` : comma-separated values (CSV) files
- `read_tsv()` : tab-separated values (TSV) files
- `read_delim()` : delimited files (CSV and TSV are important special cases)
- `read_fwf()` : fixed-width files
- `read_table()` : whitespace-separated files
- `read_log()` : web log files

https://readr.tidyverse.org/

Time series handling

STE1_TOA5_15475.Table1.dat

```
"TOA5","15475","CR1000X","15475","CR1000X.Std.04.01","CPU:STE_Site1_HRM_2020.CR1X","27479","Table1"
"TIMESTAMP","RECORD","BattV_Avg","deltaT1(1)","deltaT1(2)","deltaT1(3)","deltaT1(4)","deltaT1(5)","deltaT1(6)","deltaT2(1)","deltaT2(2)","deltaT2(3)","deltaT
,"deltaT7(1)","deltaT7(2)","deltaT7(3)","deltaT7(4)","deltaT7(5)","deltaT7(6)","deltaT8(1)","deltaT8(2)","deltaT8(3)","deltaT8(4)","deltaT8(5)","deltaT8(6)",
"TS","RN","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","","",
"","","Avg","Smp","Smp","Smp","Smp","Smp","Smp","Smp","Smp","Smp","Smp","Smp","Smp","Smp","Smp","Smp","Smp","Smp","Smp","Smp","Smp","Smp","Smp","Smp","
"2020-07-23 11:16:00",0,12.08,0.01,-0.011,-0.025,0.012,-0.007,-0.021,0.006,-0.013,-0.021,0.007,-0.011,-0.021,0,-0.011,-0.017,0.007,-0.01,-0.017,-0.009,0.003,
"2020-07-23 11:46:00",1,12.08,-0.001,-0.002,-0.008,-0.003,0.002,-0.001,0,-0.001,-0.004,0,-0.003,-0.003,-0.002,0,-0.002,-0.003,-0.033,-0.003,0.002,0.006,0.005
"2020-07-23 12:16:00",2,12.08,0.009,0.004,0,0.008,0.002,0,0.004,0.001,0.001,0.004,0,-0.001,0.006,0.001,-0.001,0.004,-0.002,0,0.006,0.006,0.004,0.007,0.004,0.
"2020-07-23 12:46:00",3,12.08,0.002,0.003,0.003,-0.003,-0.001,0,0.001,0.001,0.002,-0.001,-0.001,0,-0.003,-0.003,0,-0.003,0.01,-0.002,0.001,0.001,0.001,-0.001
"2020-07-23 13:16:00",4,12.08,-0.002,0.007,0.002,0.009,-0.001,0.001,0.002,0,0,0.002,0,0.001,0.006,0.003,0.002,0.007,0.011,0,0.002,0,-0.001,0.003,0.004,0,0.00
"2020-07-23 13:46:00",5,12.08,0.008,0.004,0.002,0.006,0.001,0.003,0.005,0.002,0.003,0.003,0.002,0.002,0.004,0.004,0.003,0.005,0.025,0.004,0.004,0.002,0.001,-
"2020-07-23 14:16:00",6,12.08,0.379,0.965,0.894,0.308,0.917,0.933,0.354,0.609,0.577,0.263,0.618,0.582,0.355,0.486,0.448,0.217,0.275,0.435,0.004,0.001,0.007,0
"2020-07-23 14:46:00",7,12.09,0.054,0.058,0.085,0.027,0.041,0.073,0.254,0.073,0.081,0.114,0.051,0.073,0.115,0.049,0.071,0.054,0.041,0.069,-0.003,-0.001,-0.00
"2020-07-23 15:16:00",8,12.08,-0.172,0.013,0.026,-0.082,0.012,0.022,0.319,0.028,0.031,0.149,0.005,0.031,0.021,0.04,0.034,0.01,0.038,0.026,-0.006,-0.003,-0.00
"2020-07-23 15:46:00",9,12.09,0.028,0.051,0.019,0.006,0.027,0.02,-0.17,-0.026,0.026,-0.017,0.012,0.02,-0.28,-0.018,0.015,-0.121,0.003,0.01,-0.001,0,-0.003,-0
"2020-07-23 16:16:00",10,12.09,-0.124,0,0.009,-0.052,0.009,0.005,-0.006,0.043,0.014,-0.057,0.015,0.005,0.157,0.04,0.011,0.092,0.007,0.01,-0.003,-0.001,-0.002
"2020-07-23 16:46:00",11,12.09,-0.049,-0.062,0,-0.017,-0.027,0.001,0.47,-0.008,-0.003,0.2,-0.028,0.001,0.033,0.007,0.009,0.004,-0.007,0.007,0.002,0.002,0.001
"2020-07-23 17:16:00",12,12.09,-0.546,-0.028,0.009,-0.263,0.001,0.004,-0.505,-0.115,0.007,-0.24,-0.029,0.01,-0.08,-0.046,0.007,-0.031,0.007,0.01,0.007,-0.001
"2020-07-23 17:46:00",13,12.09,-0.852,-0.01,-0.001,-0.447,0.031,-0.015,0.562,-0.039,0.001,0.272,-0.038,0.005,0.034,0.01,0.007,0.009,0.004,0.007,0.001,0,0.001
"2020-07-23 18:16:00",14,12.09,0.009,0.064,0.004,-0.011,0.034,0.004,0.056,0.072,0.013,-0.026,0.035,0.022,0.075,0.018,0.021,0.06,0.024,0.006,-0.001,-0.001,-0.
01,0.001,-0.00
004,0.002,0.00
,-0.004,0.01,-
,0.001,-0.002,
0,-0.003,-0.00
006,-0.003,-0.
```

	TIMESTAMP	RECORD	BattV_Avg	deltaT1(1)	deltaT1(2)	deltaT1(3)	deltaT1(4)	deltaT1(5)	deltaT1(6)	deltaT2(1)	deltaT2(2)	deltaT2(3)	deltaT2(4)
1	2020-07-23 11:16:00	0	12.08	0.010	-0.011	-0.025	0.012	-0.007	-0.021	0.006	-0.013	-0.021	0.007
2	2020-07-23 11:46:00	1	12.08	-0.001	-0.002	-0.008	-0.003	0.002	-0.001	0.000	-0.001	-0.004	0.000
3	2020-07-23 12:16:00	2	12.08	0.009	0.004	0.000	0.008	0.002	0.000	0.004	0.001	0.001	0.004
4	2020-07-23 12:46:00	3	12.08	0.002	0.003	0.003	-0.003	-0.001	0.000	0.001	0.001	0.002	-0.001
5	2020-07-23 13:16:00	4	12.08	-0.002	0.007	0.002	0.009	-0.001	0.001	0.002	0.000	0.000	0.002
6	2020-07-23 13:46:00	5	12.08	0.008	0.004	0.002	0.006	0.001	0.003	0.005	0.002	0.003	0.003
7	2020-07-23 14:16:00	6	12.08	0.379	0.965	0.894	0.308	0.917	0.933	0.354	0.609	0.577	0.263
8	2020-07-23 14:46:00	7	12.09	0.054	0.058	0.085	0.027	0.041	0.073	0.254	0.073	0.081	0.114
9	2020-07-23 15:16:00	8	12.08	-0.172	0.013	0.026	-0.082	0.012	0.022	0.319	0.028	0.031	0.149
10	2020-07-23 15:46:00	9	12.09	0.028	0.051	0.019	0.006	0.027	0.020	-0.170	-0.026	0.026	-0.017
11	2020-07-23 16:16:00	10	12.09	-0.124	0.000	0.009	-0.052	0.009	0.005	-0.006	0.043	0.014	-0.057
12	2020-07-23 16:46:00	11	12.09	-0.049	-0.062	0.000	-0.017	-0.027	0.001	0.470	-0.008	-0.003	0.200

```
data = read.table("datafile.dat",
                  header=F,
                  sep=",",
                  skip=4,
                  na.strings = "NAN")

data$TIMESTAMP = ymd_hms(data$TIMESTAMP, tz="America/Toronto")
```

Time series handling

Dates

- time stamps

https://rawgit.com/rstudio/cheatsheets/main/lubridate.pdf

2017-11-28T14:02:00	**ymd_hms(), ymd_hm(), ymd_h().** ymd_hms("2017-11-28T14:02:00")
2017-22-12 10:00:00	**ydm_hms(), ydm_hm(), ydm_h().** ydm_hms("2017-22-12 10:00:00")
11/28/2017 1:02:03	**mdy_hms(), mdy_hm(), mdy_h().** mdy_hms("11/28/2017 1:02:03")
1 Jan 2017 23:59:59	**dmy_hms(), dmy_hm(), dmy_h().** dmy_hms("1 Jan 2017 23:59:59")
20170131	**ymd(), ydm().** ymd(20170131)
July 4th, 2000	**mdy(), myd().** mdy("July 4th, 2000")
4th of July '99	**dmy(), dym().** dmy("4th of July '99")

2018-01-31 11:59:59	**day(x)** Day of month. day(dt) **wday(x, label, abbr)** Day of week. **qday(x)** Day of quarter.
2018-01-31 11:59:59	**hour(x)** Hour. hour(dt)
2018-01-31 11:59:59	**minute(x)** Minutes. minute(dt)
2018-01-31 11:59:59	**second(x)** Seconds. second(dt)

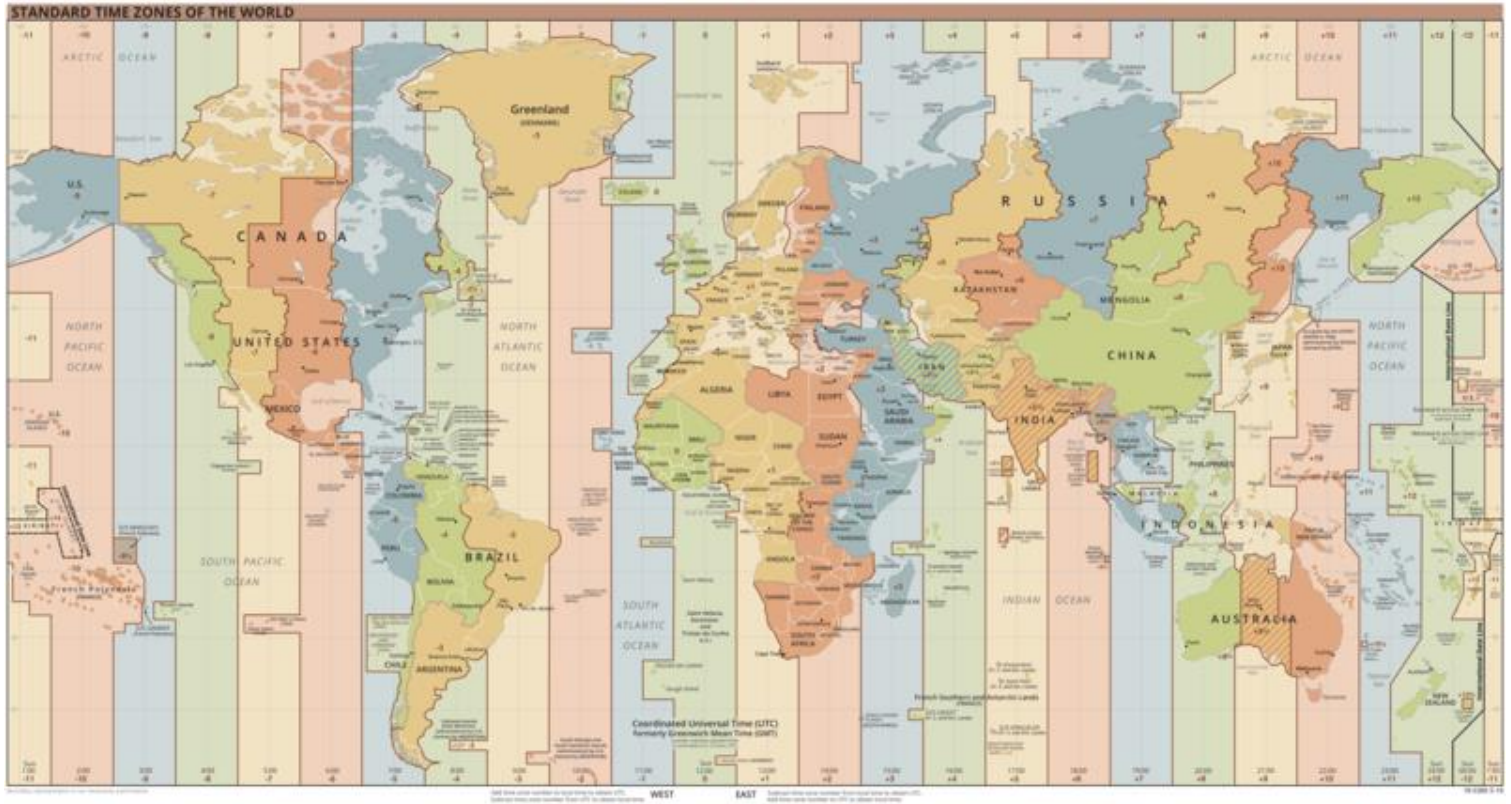
More examples

- base R: https://deep-tools.netlify.app/docs-workshops/uhelsinki-workshop2021/01_datacleanr/
- lubridate: https://lubridate.tidyverse.org/

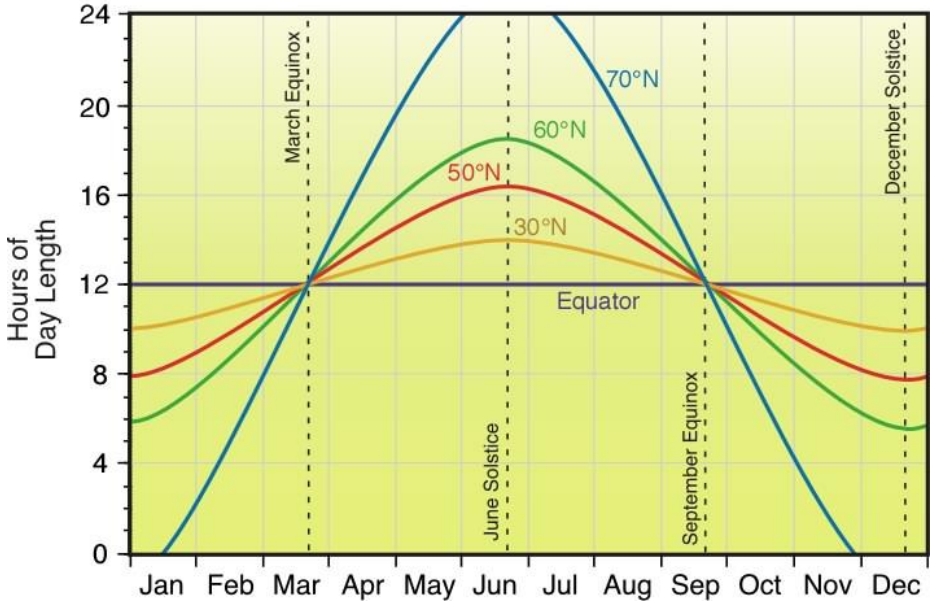
Time series handling

Dates

- time zones (daylight savings ?)
- `solartime` (https://github.com/bgctw/solartime; noon when sun is at zenith)



https://commons.wikimedia.org/wiki/File:World_Time_Zones_Map.png

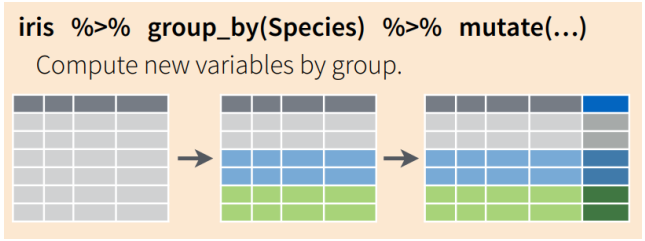


http://www.physicalgeography.net/fundamentals/6i.html

Time series handling

Temporal aggregation

- base R
- `dplyr`



30-min data

site	sensor	tree	depth	timestamp	time	tz	value
STE1	deltaT4_out	4	out	2020-08-24 18:46:00	2020-08-24 18:46:00	EDT	-0.025268594
STE1	deltaT4_out	4	out	2020-08-24 19:16:00	2020-08-24 19:16:00	EDT	-0.027292142
STE1	deltaT4_out	4	out	2020-08-24 19:46:00	2020-08-24 19:46:00	EDT	-0.025367214
STE1	deltaT4_out	4	out	2020-08-24 20:16:00	2020-08-24 20:16:00	EDT	-0.029156584
STE1	deltaT4_out	4	out	2020-08-24 20:46:00	2020-08-24 20:46:00	EDT	-0.029327615
STE1	deltaT4_out	4	out	2020-08-24 21:16:00	2020-08-24 21:16:00	EDT	-0.031191612
STE1	deltaT4_out	4	out	2020-08-24 21:46:00	2020-08-24 21:46:00	EDT	-0.031252544

1-h data

sensor	site	date	doy	hour	tree	depth	tz	value
deltaT4_out	STE1	2020-08-24 17:00:00	237	17	4	out	EDT	-0.0232645516
deltaT4_out	STE1	2020-08-24 18:00:00	237	18	4	out	EDT	-0.0223245229
deltaT4_out	STE1	2020-08-24 19:00:00	237	19	4	out	EDT	-0.0263296781
deltaT4_out	STE1	2020-08-24 20:00:00	237	20	4	out	EDT	-0.0292420997

```r
# aggregate to hourly data
data_h = data %>%
  group_by(sensor, site,
           date=ymd_h(substr(data$timestamp, 1, 13)),
           doy=yday(data$timestamp),
           hour=hour(data$timestamp)) %>%
  summarise(tree = unique(tree),
            depth=unique(depth),
            tz=unique(tz),
            value=mean(value, na.rm=T))
```

More examples

- aggregate(·) https://deep-tools.netlify.app/talk/uhelsinki-2021-rpeters-ahurley/
- dplyr | group_by(·) https://www.rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf

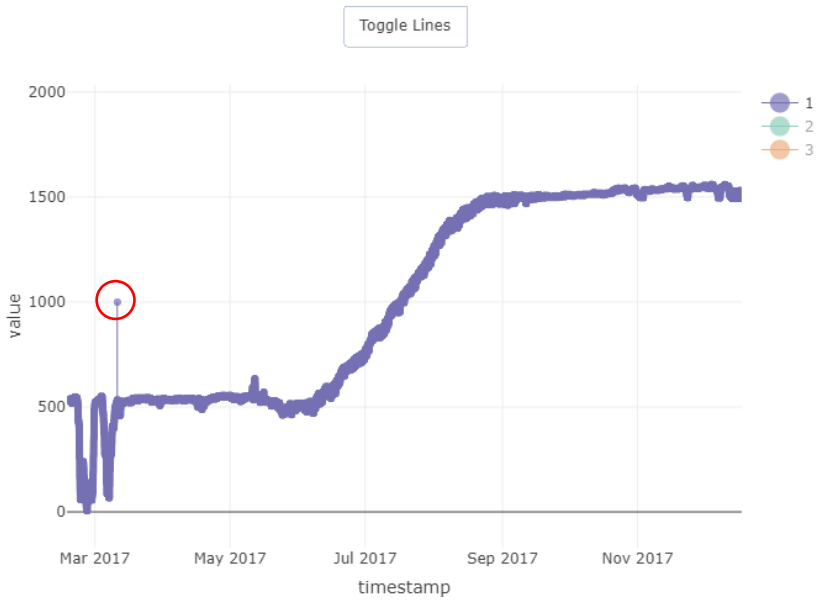
Data ‘cleaning’

Goal? *Interactive and reproducible data inspection, visualization and cleaning.*

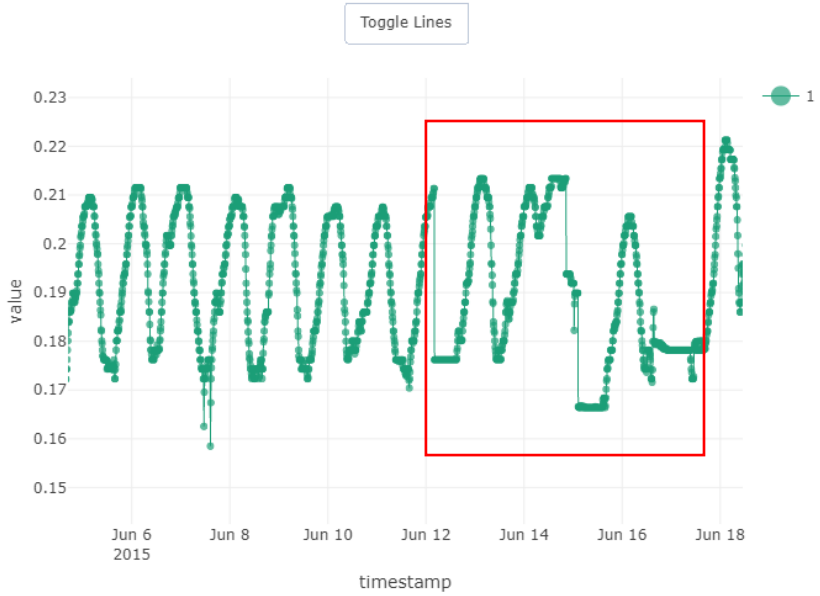
Typical data issues

Outlier and sensor failure issues *Removing data should always be done with care!*

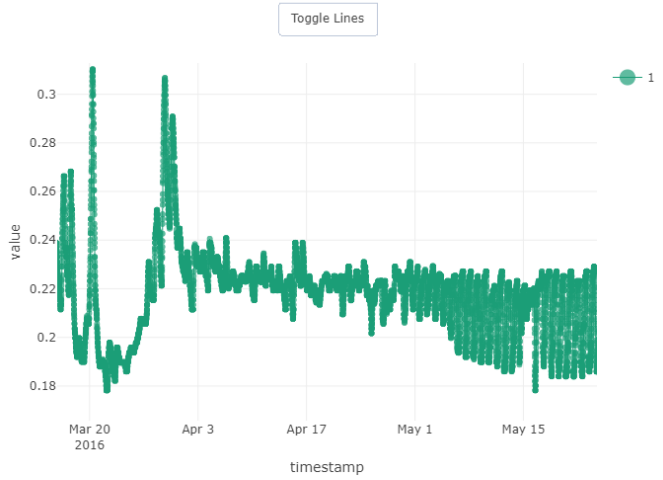
Outliers (dendrometer data)

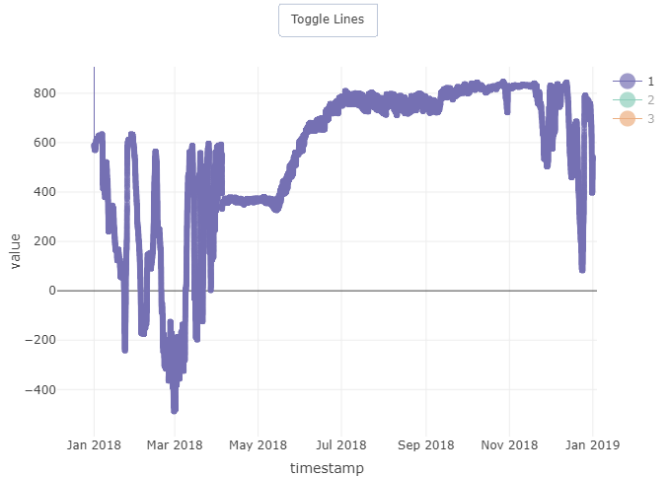


Sensor failure (sap flow data)



What about spring data?





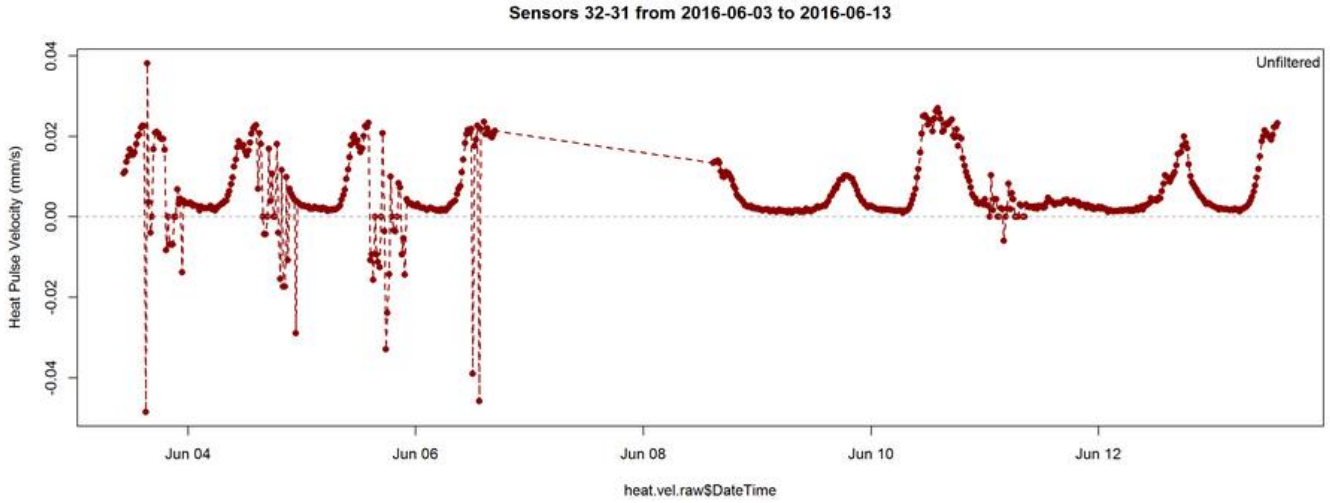
Data ‘cleaning’ – ways forward?

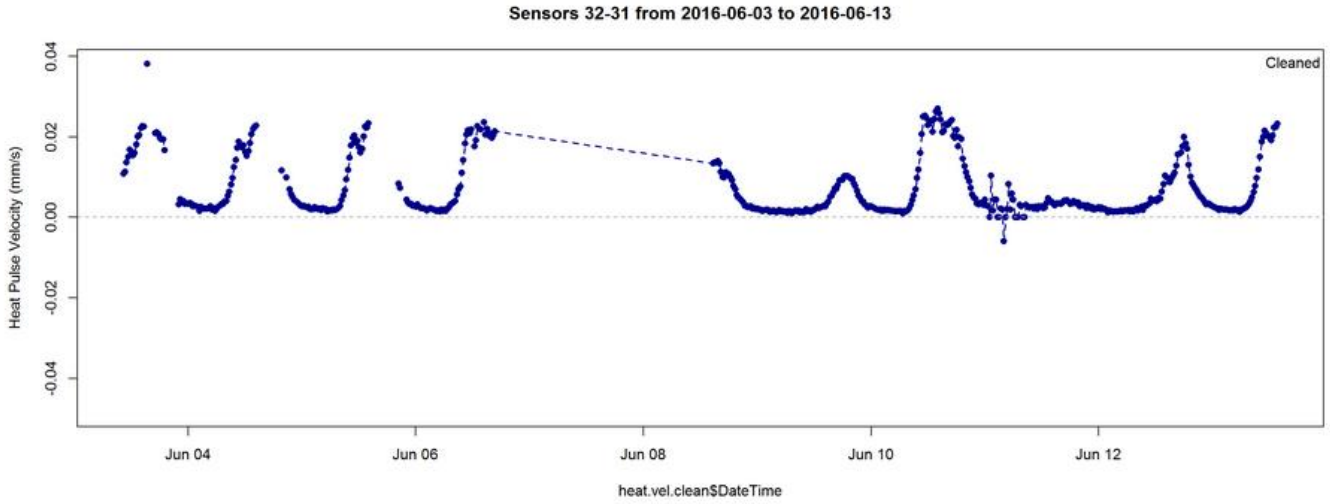
Manual processing:

- *Quality Control with diagnostic plots*
  - *Time Series*
  - *X/Y (e.g., Sapflow vs. Environment)*

- *Quality Assurance with code:*
  - *Filters*
    - *Moving average*
    - *Threshold*
    - *Outlier detection*
  - *Adjustments*

→ *Long series & many sites = much time!*
→ *No “one-size-fits-all” solutions*
→ ***Must be done reproducibly***

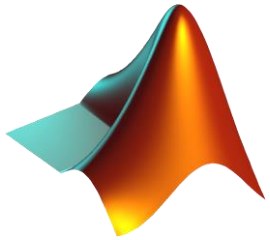




Data ‘cleaning’ – ways forward?

(Reproducible) processing and cleaning:

- *Several excellent **programmatic** tools*
  - → *Spreadsheet software (!)*
  - → *Software & libraries*
  - → *Bespoke code*



Focus on R and packages!

- *Extensively used in Environmental Sciences*
  - → *interoperable*

Long series, many sites.. ?

Data ‘cleaning’ with `datacleanr`

ESACSEE

A new R-based package

A flexible and efficient tool for interactive data cleaning


PLOS ONE

PUBLISH ABOUT BROWSE

OPEN ACCESS PEER-REVIEWED

RESEARCH ARTICLE

Addressing the need for interactive, efficient, and reproducible data processing in ecology with the datacleanr R package

Alexander G. Hurley, Richard L. Peters, Christoforos Pappas, David N. Steger, Ingo Heinrich

Published: May 12, 2022 • https://doi.org/10.1371/journal.pone.0268426

```
# grab all data from ZHANG
zhang <- cosore::csr_table("data", c("d20190424_ZHANG_maple",
                                     "d20190424_ZHANG_oak")) %>%
  # adjust for grouping
  mutate(CSR_PORT = as.factor(CSR_PORT))

# group by CSR_DATASET and CSR_PORT
datacleanr::dcr_app(zhang)
```

Properties:

- Uses R (links with other packages);
- Freely available (avoid license costs);
- Uses R shiny (interactive approach).
- **Reproducible**

Structure of the tool:

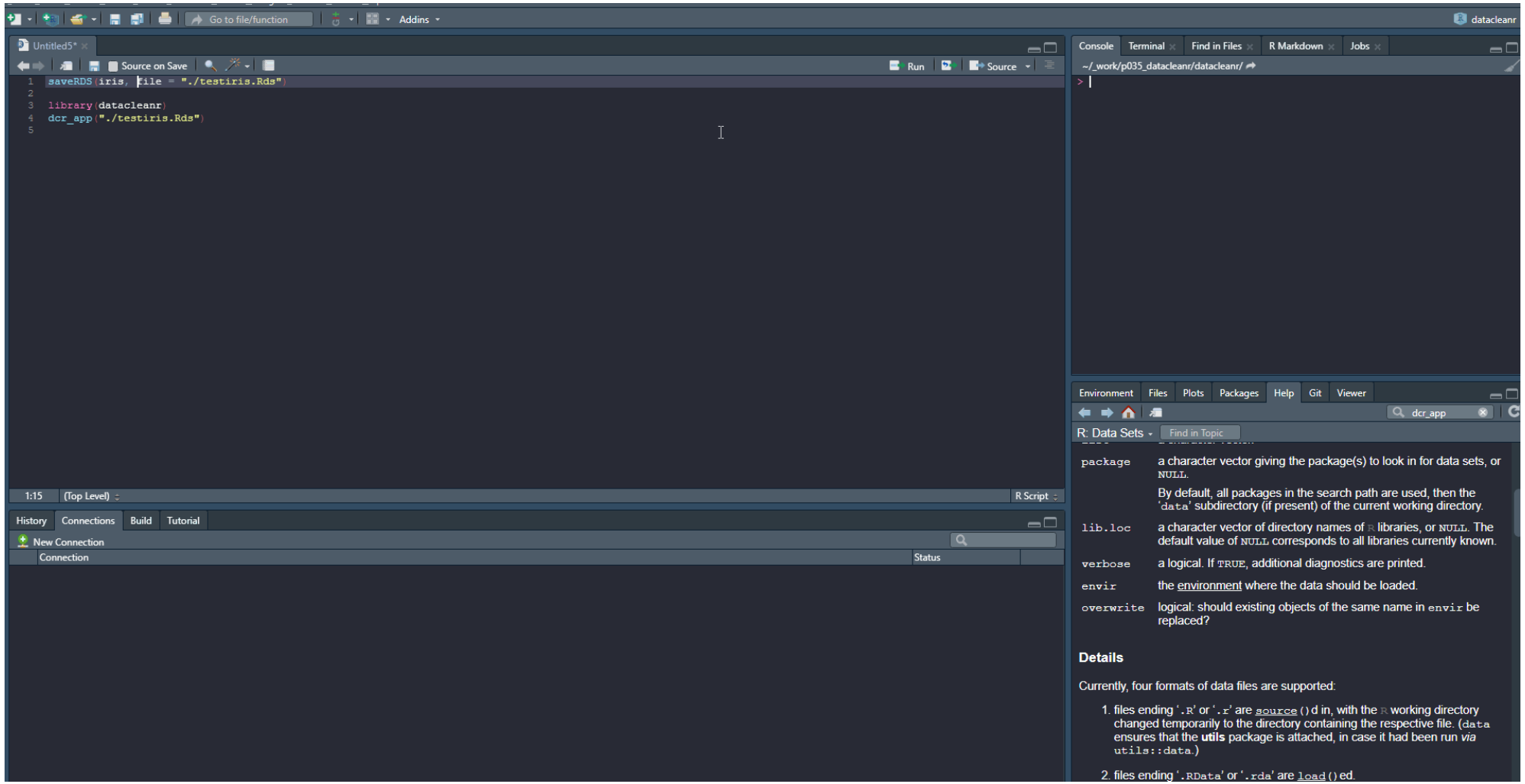
- Set-up & overview;
- Filtering;
- Visual cleaning and annotating;
- Extract.

Data ‘cleaning’ with `datacleanr`

ESACSEE

Versatile

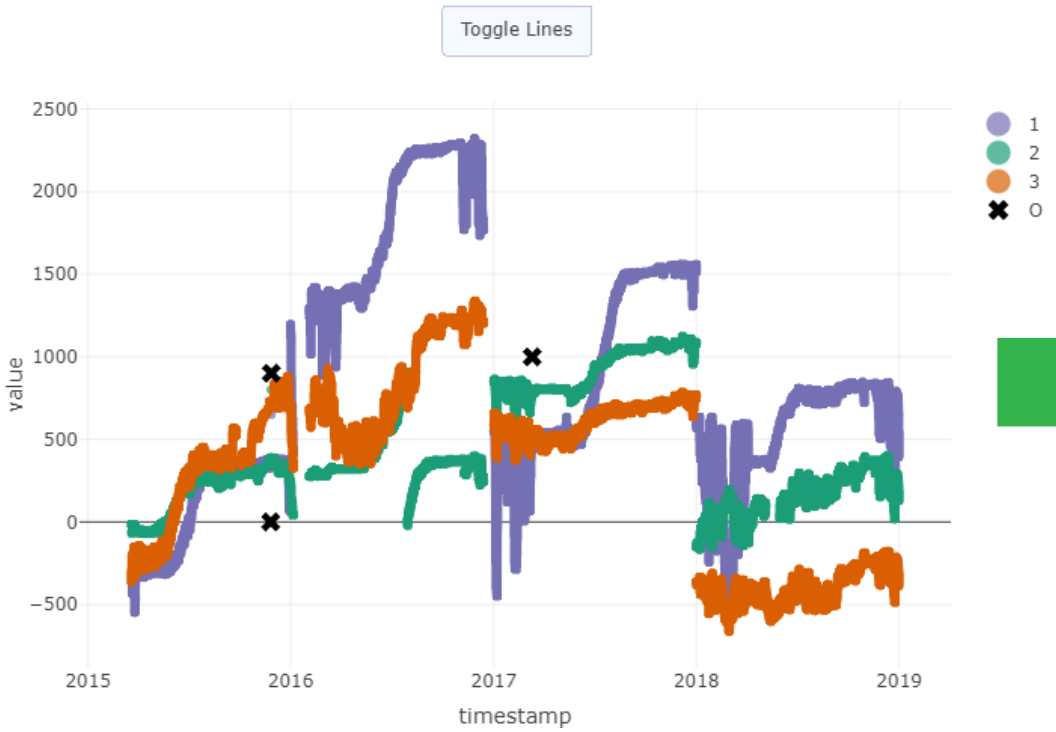
Tested on multiple temporal-and spatial-specific data sets

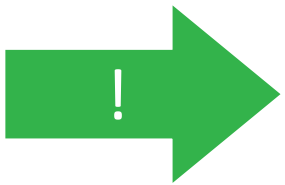


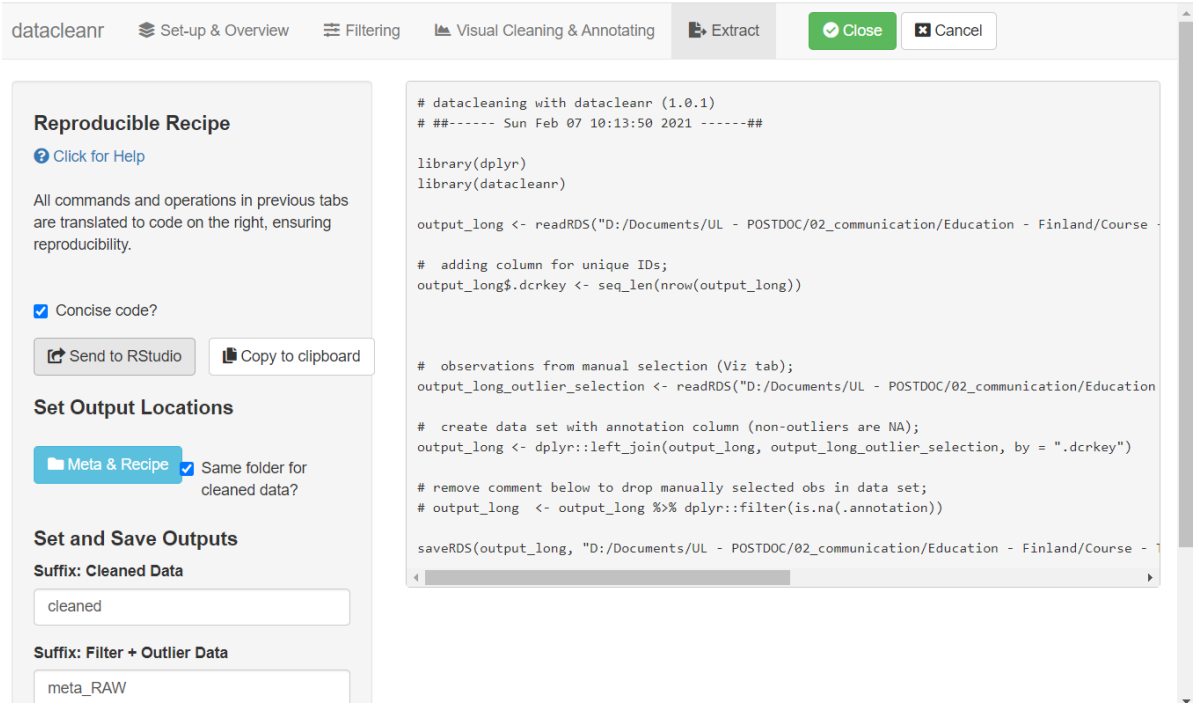
Data ‘cleaning’ with `datacleanr`

Extract

Reproducible recipe to cook up some fresh data







Additional Resources

Hurley AG, Peters RL, Pappas C, Steger DN, Heinrich I (2022)
Addressing the need for interactive, efficient, and reproducible data processing in ecology with the datacleanr R package. PLoS ONE 17(5): e0268426. https://doi.org/10.1371/journal.pone.0268426

https://github.com/the-hull/datacleanr

https://deep-tools.netlify.app/#workshops

Data processing

`TREX` `treenetproc`

Goal? *Make standardized time series processing workflows for widely used monitoring techniques more accessible.*

Sap flow measurements

TREX

Sap flow sensors - Utilizing the TREX R package (https://the-hull.github.io/TREX)

TREX 1.0.0 | Reference

Médiane | Gisèle Trudel @UQAM

Peters *et al.* (2020)

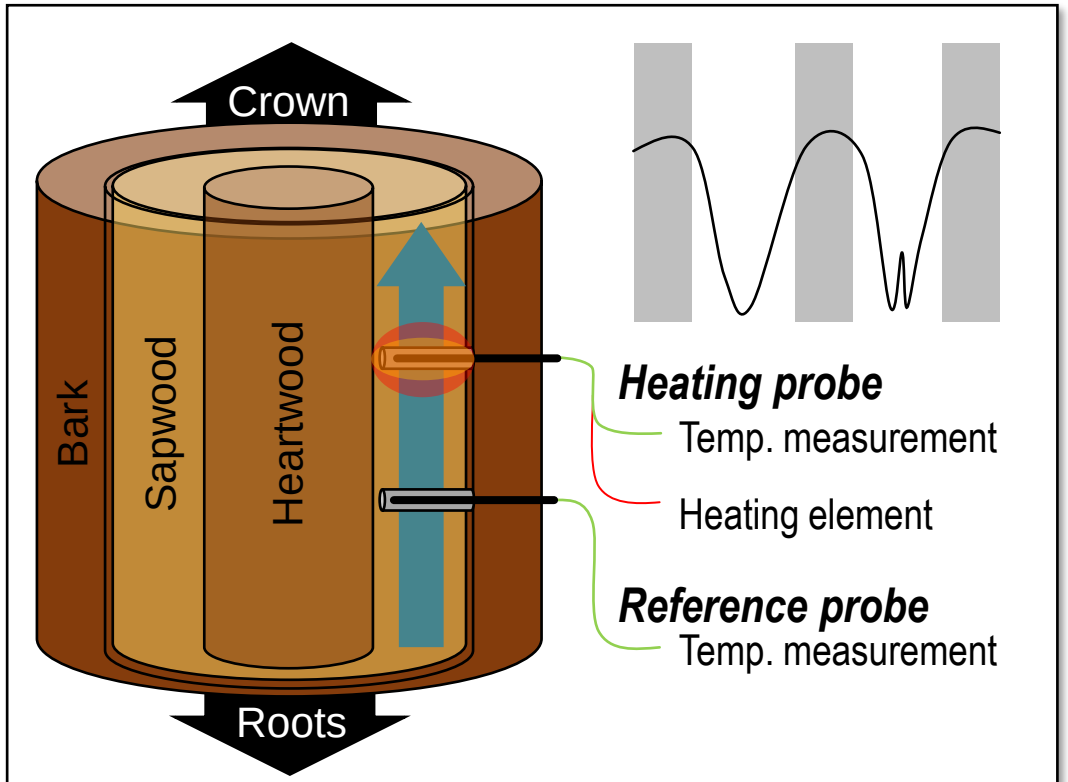


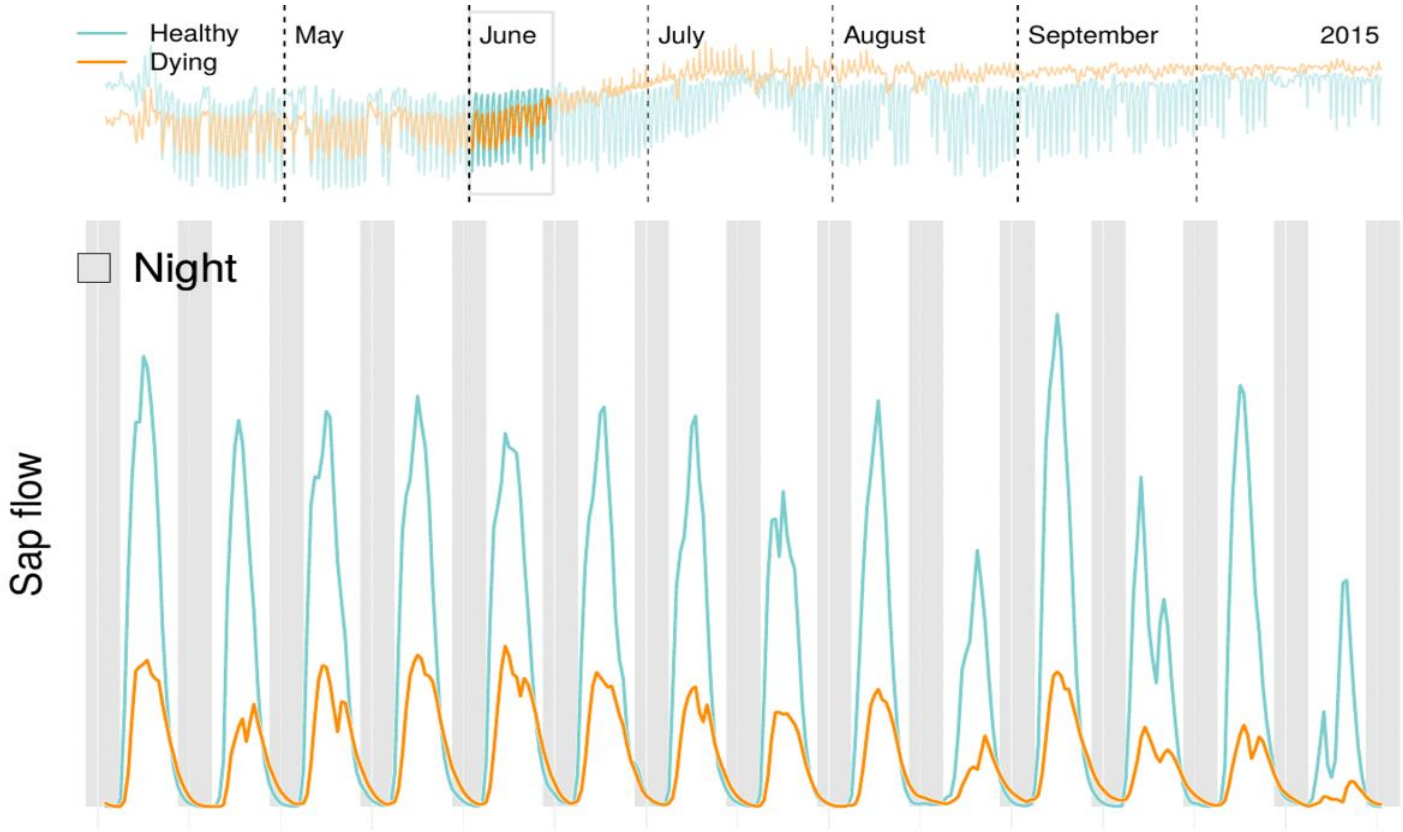
Fig. Schematics of thermal dissipation method.

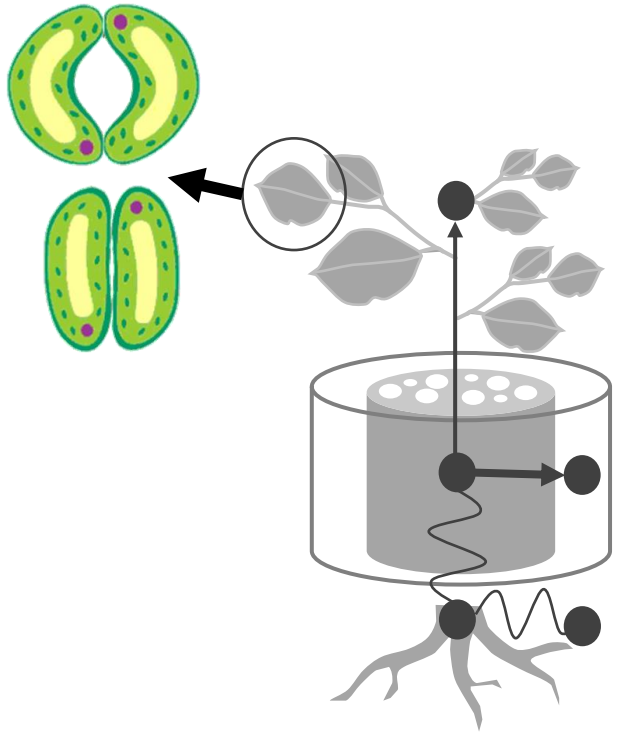
Sap flow measurements

TREX

ESACSEE

Input – Thermal dissipation probe measurements
tdm.data()



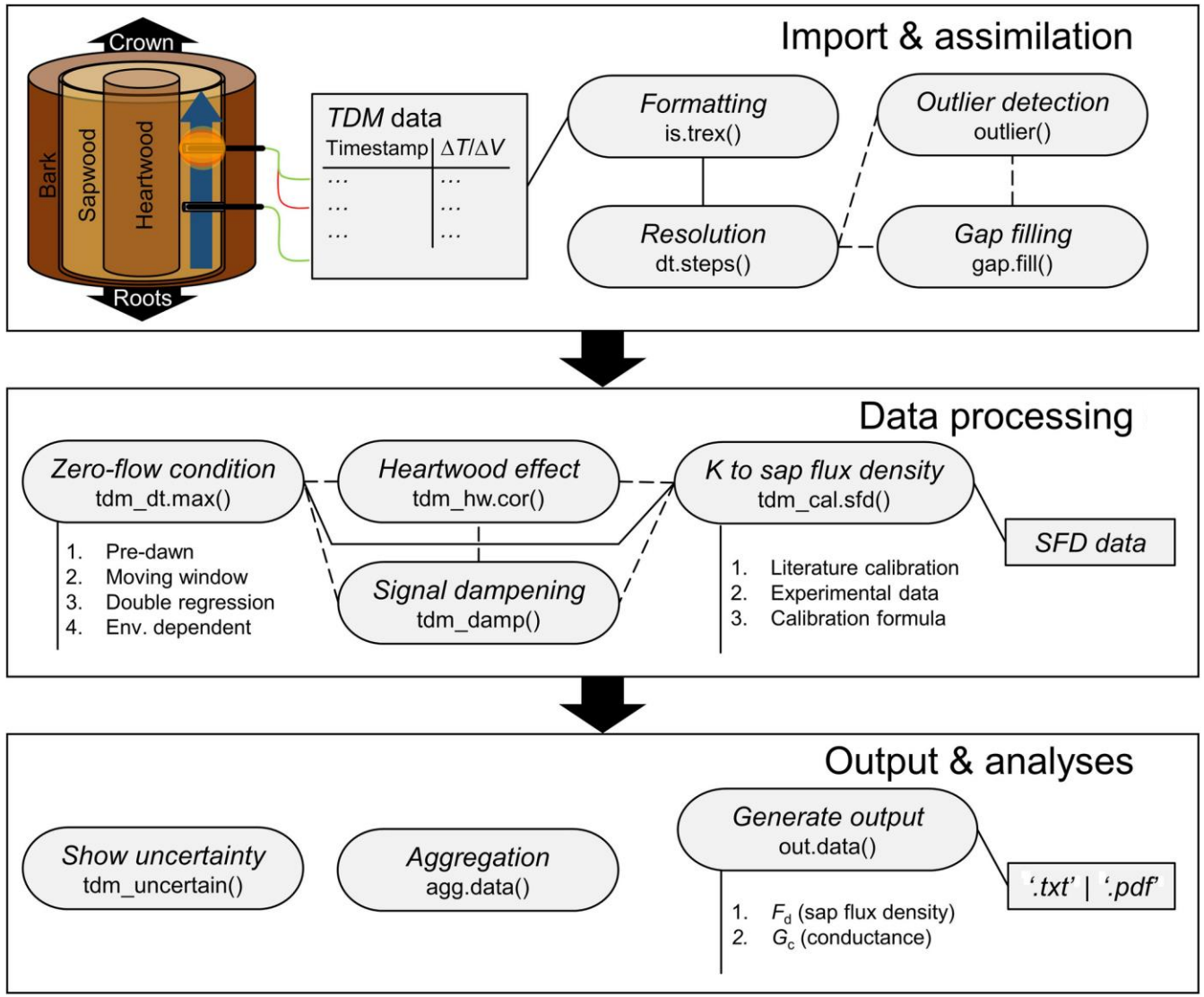


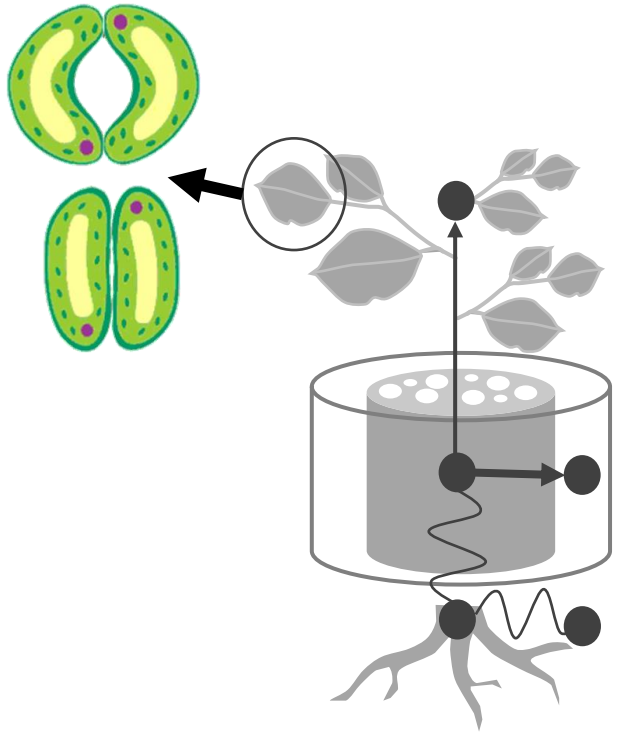
Sap flow measurements

TREX

ESACSEE

Workflow



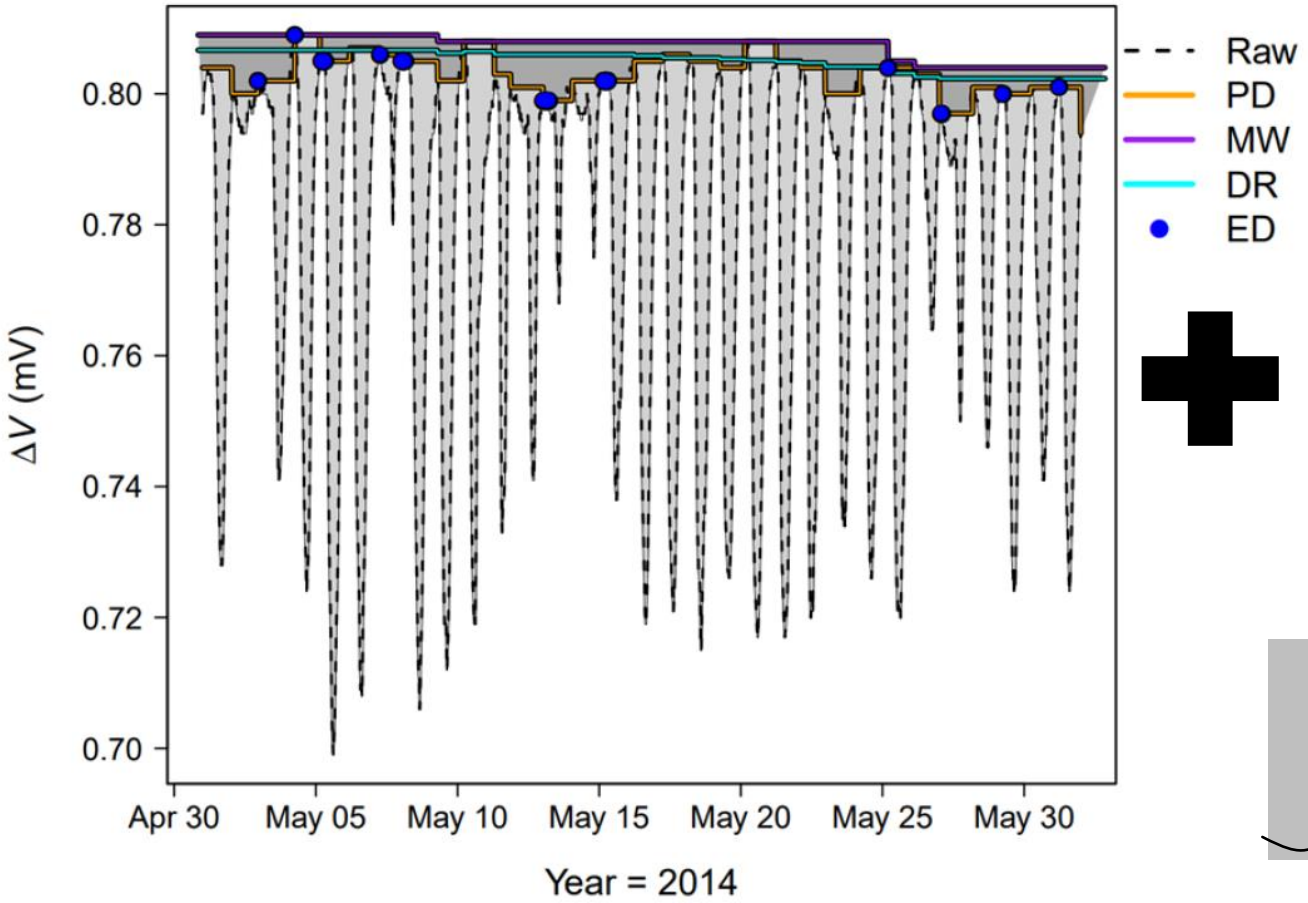


Sap flow measurements

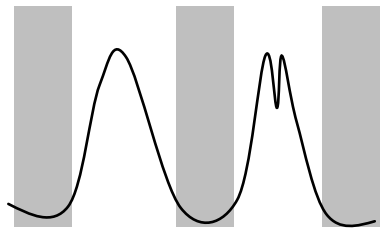
TREX

ESACSEE

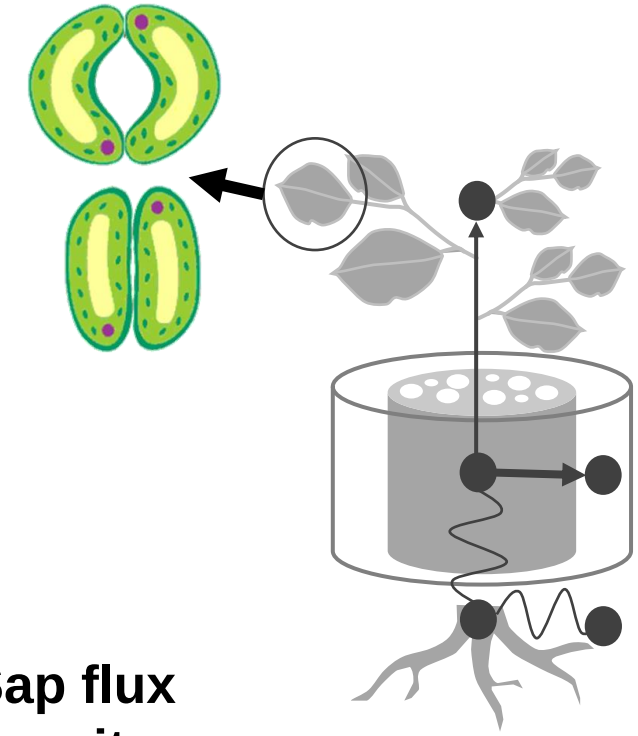
Functionalities – Determining zero flow
tdm_dt.max()



Calibration values
tdm_cal.sfd()



Sap flux density

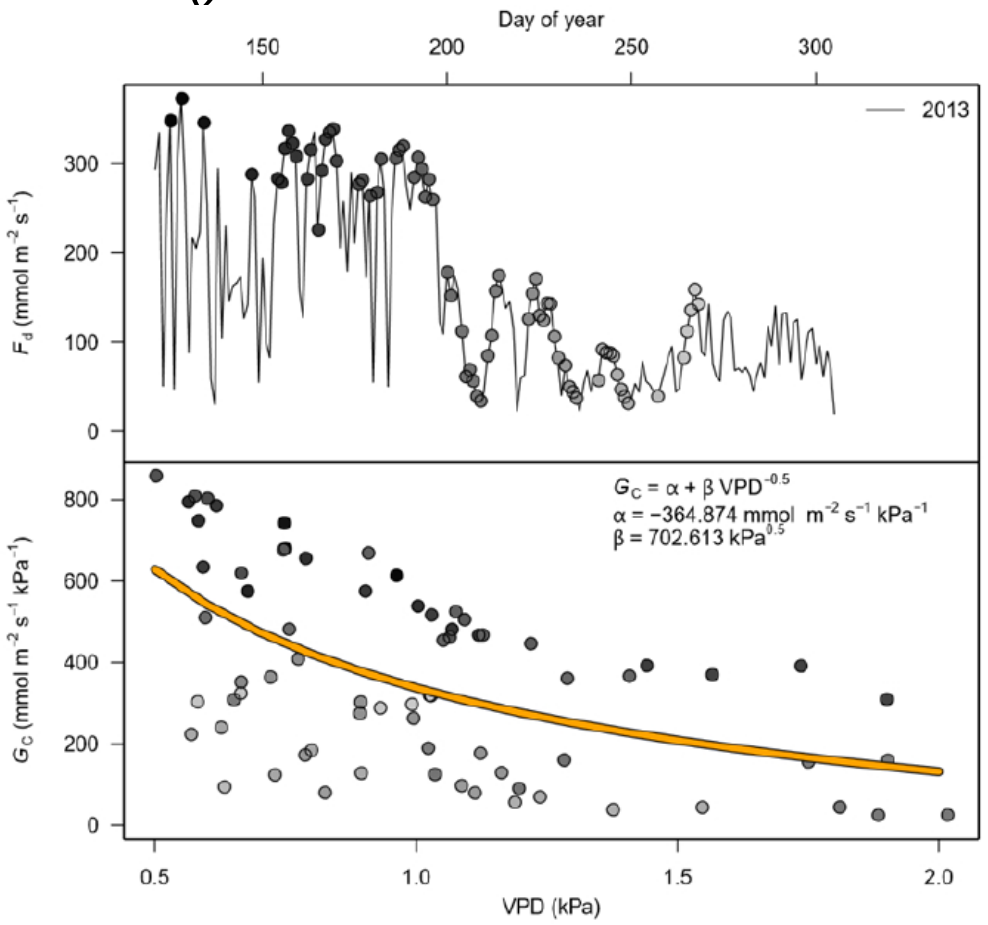


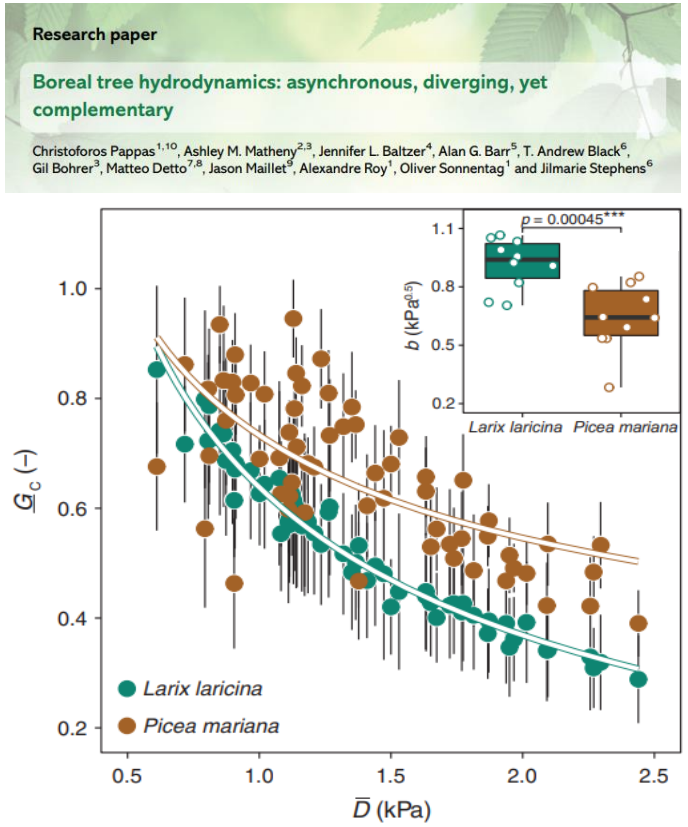
Sap flow measurements

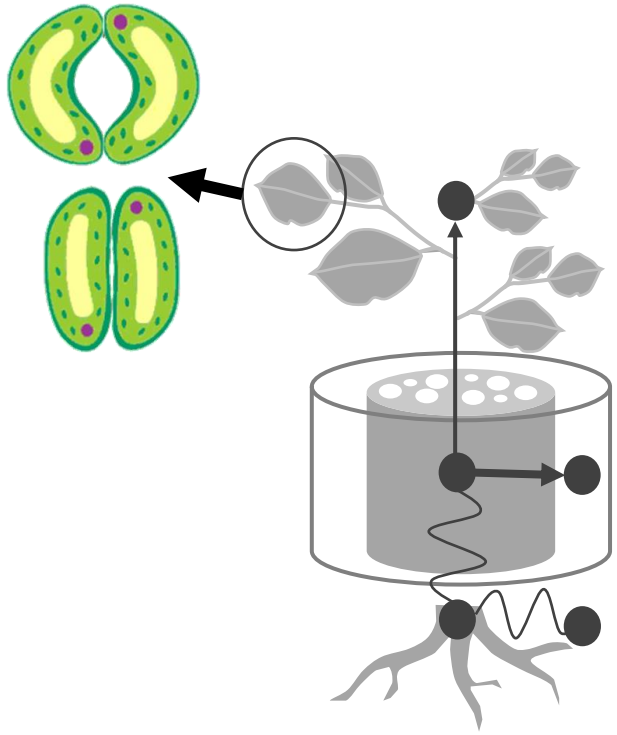
TREX

ESACSEE

Output – Canopy conductance behavior
out.data()







Additional Resources

Peters, RL, Pappas, C, Hurley, AG, et al. (2021) **Assimilate, process and analyse thermal dissipation sap flow data using the TREX r package.** Methods Ecol Evol. https://doi.org/10.1111/2041-210X.13524

https://the-hull.github.io/TREX/index.html

https://deep-tools.netlify.app/#workshops

Dendrometer measurements

treenetproc

Dendrometers - Utilizing the treenetproc R package (https://github.com/treenet/treenetproc)

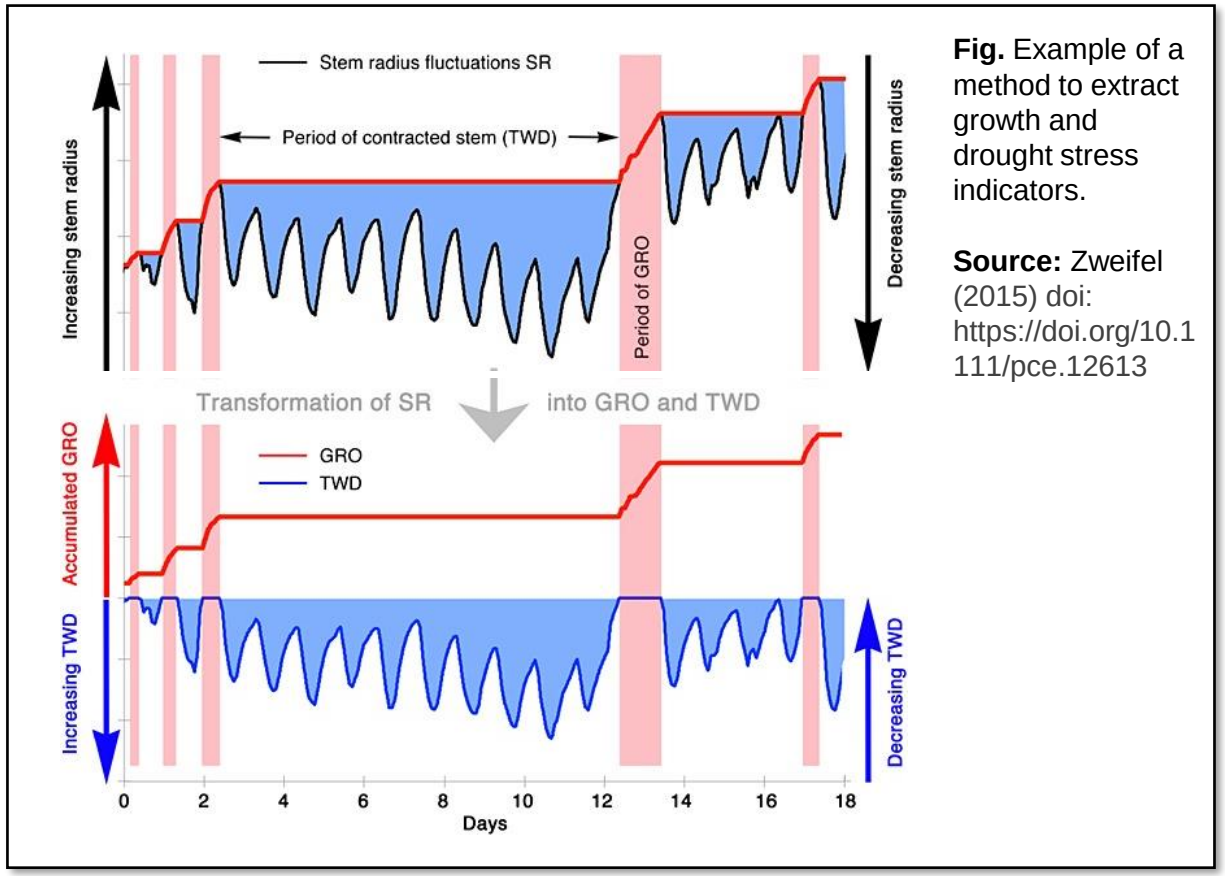


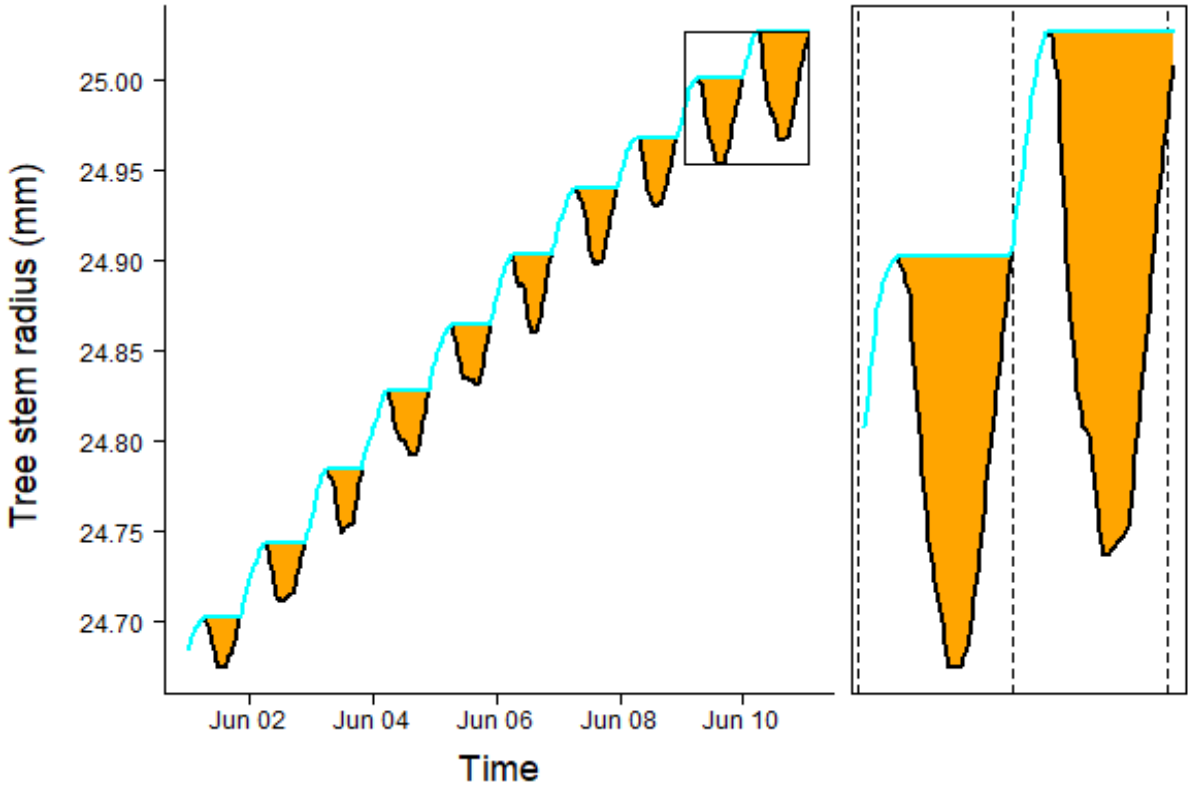
Fig. Example of a method to extract growth and drought stress indicators.

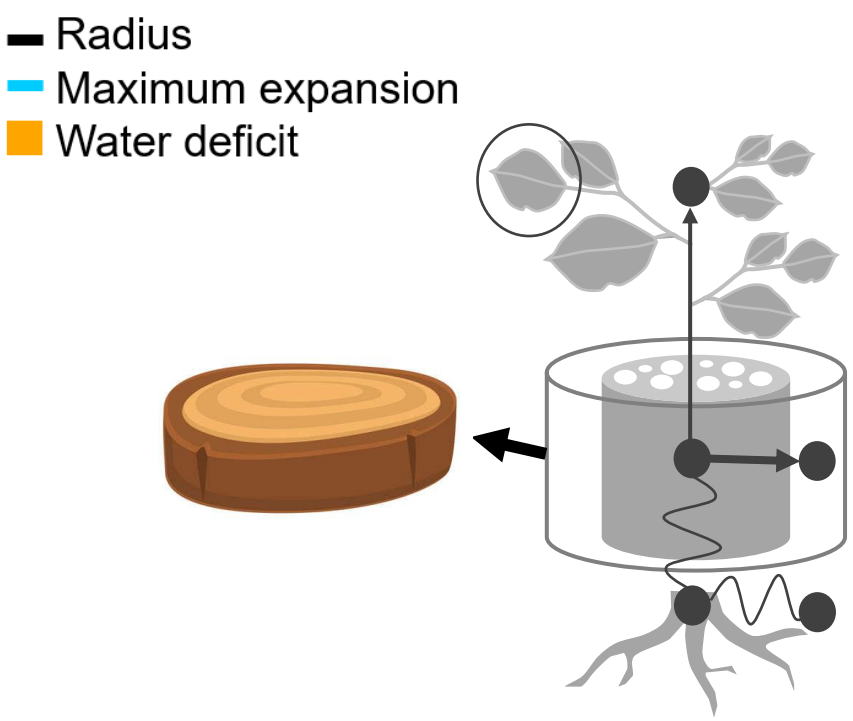
Source: Zweifel (2015) doi: https://doi.org/10.1111/pce.12613

Dendrometer measurements

treenetproc

Input – Point dendrometer measurements
dendro_data_L0()

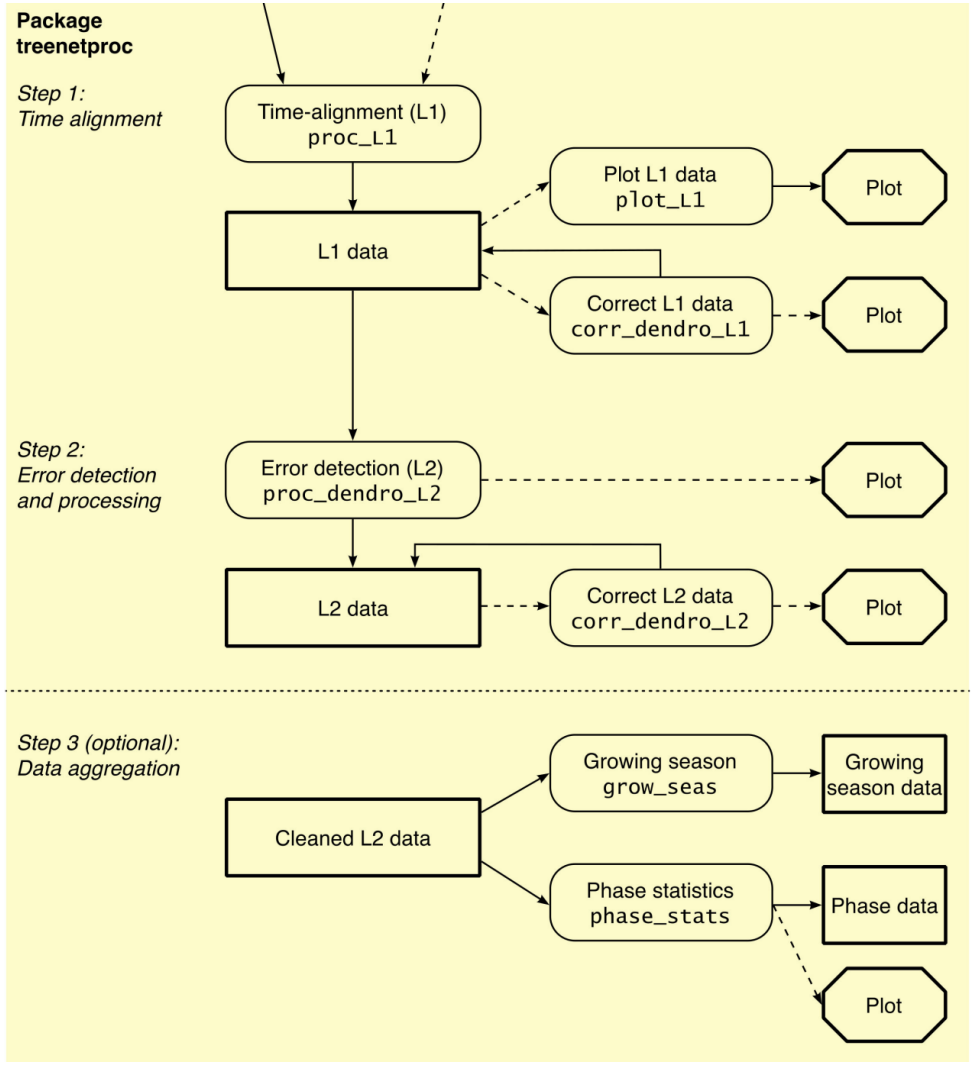


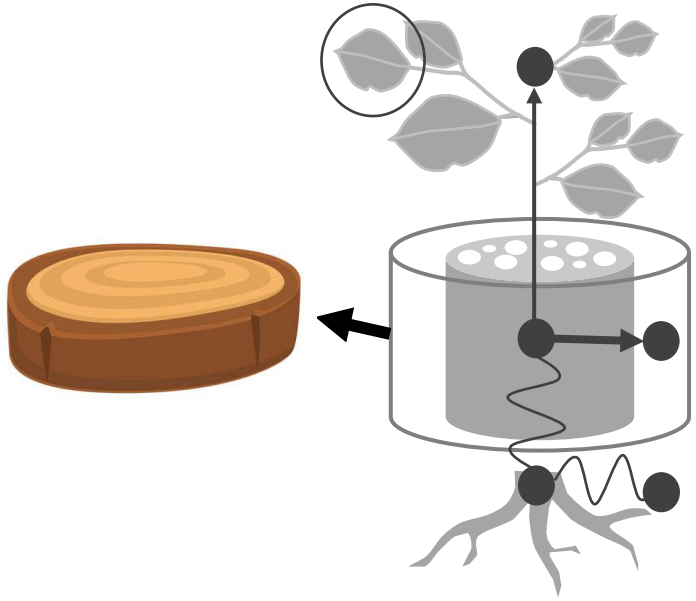


Dendrometer measurements

treenetproc

Workflow



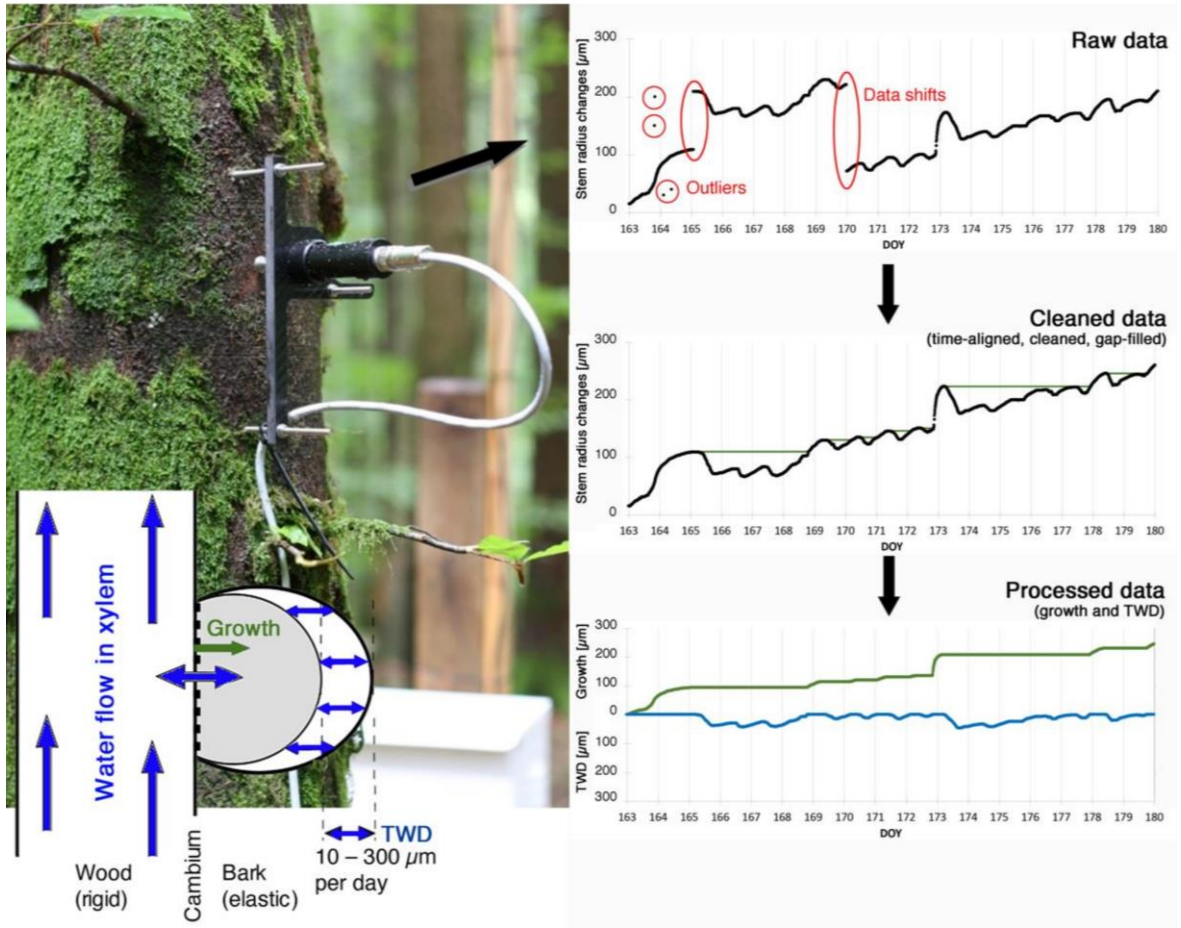


Dendrometer measurements

treenetproc

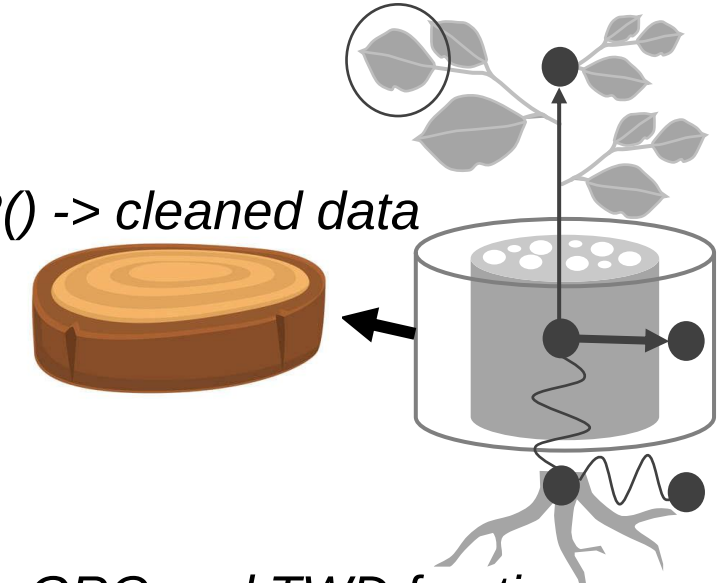
Functionalities – Extracting growth and TWD

proc_dendro_L2()



proc_dendro_L1() -> Time aligned data

proc_dendro_L2() -> cleaned data



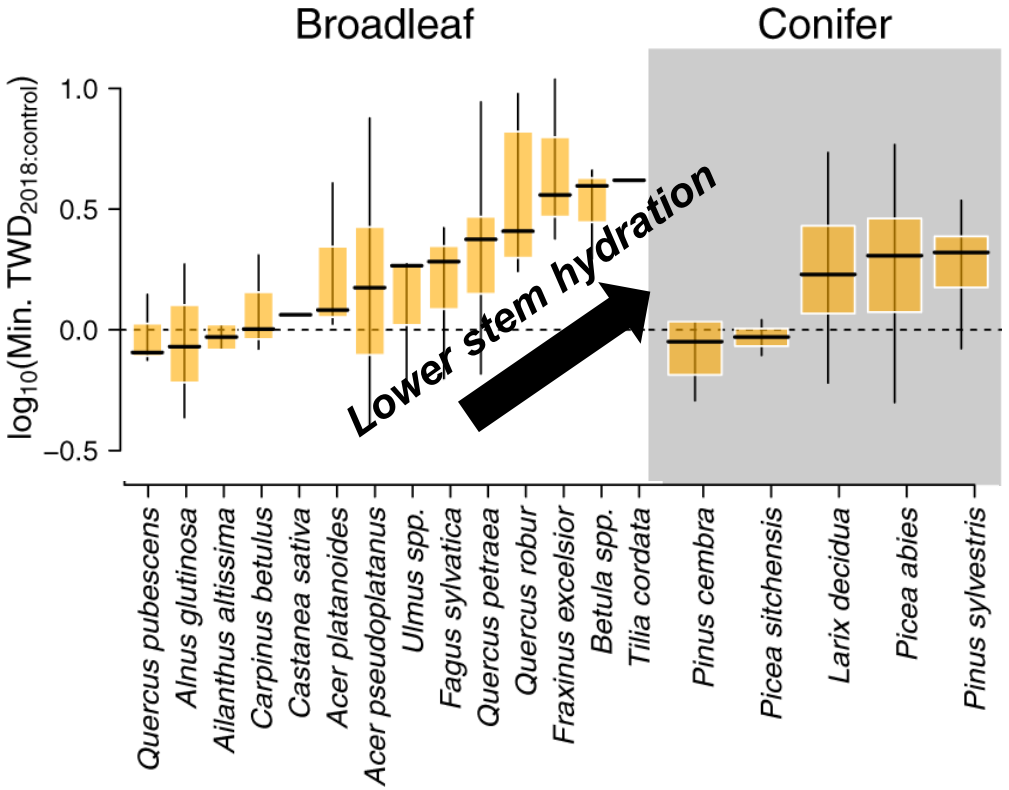
-> separated into GRO and TWD fractions

Dendrometer measurements

treenetproc

Output – Night-time tree water deficit as indicator for drought stress

proc_dendro_L2()
gro_seas
gro_stats

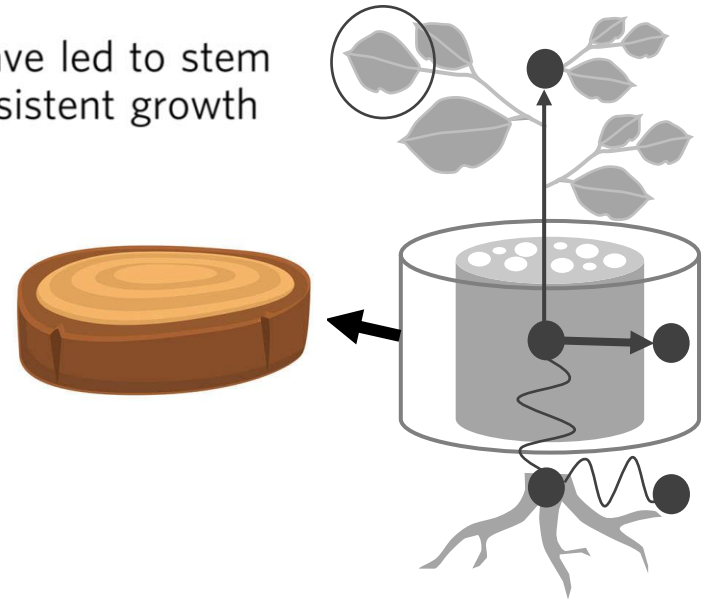


nature COMMUNICATIONS

ARTICLE

https://doi.org/10.1038/s41467-021-27579-9 OPEN

The 2018 European heatwave led to stem dehydration but not to consistent growth reductions in forests



Additional Resources

Knüsel S., Peters R.L., Haeni M., Wilhelm M., Zweifel R. (2021) **Processing and extraction of seasonal tree physiological parameters from stem radius time series.** Forests 12(6) https://doi.org/10.3390/f12060765

https://github.com/treenet/treenetproc

https://deep-tools.netlify.app/#workshops

Hands-on & Q/A