

# Package ‘neonSoilFlux’

July 22, 2025

**Type** Package

**Title** Compute Soil Carbon Fluxes for the National Ecological  
Observatory Network Sites

**Version** 1.0.0

**Description** Acquires and synthesizes soil carbon fluxes at sites located in the National Ecological Observatory Network (NEON). Provides flux estimates and associated uncertainty as well as key environmental measurements (soil water, temperature, CO<sub>2</sub> concentration) that are used to compute soil fluxes.

**URL** <https://github.com/jmzobitz/neonSoilFlux>

**BugReports** <https://github.com/jmzobitz/neonSoilFlux/issues>

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.1

**Depends** R (>= 2.10)

**Imports** dplyr, ggplot2, lubridate, neonUtilities, purrr, stats,  
stringr, tibble, tidyr, tidyselect

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** John Zobitz [aut, cre] (ORCID: <<https://orcid.org/0000-0002-1830-143X>>),  
Edward Ayres [aut] (ORCID: <<https://orcid.org/0000-0003-1846-1473>>),  
Katie O'Rourke [ctb] (LinkedIn,  
<<https://www.linkedin.com/in/katie-o-rourke-5a898b194/>>),  
Zoey Werbin [ctb],  
Lajntxiag Lee [ctb],  
Ridwan Abdi [ctb] (LinkedIn,  
<<https://www.linkedin.com/in/ridwaan-cabdi/>>),  
Dijonë Mehmeti [ctb] (LinkedIn,  
<<https://www.linkedin.com/in/dijon%C3%AB-mehmeti/>>),  
Ly Xiong [ctb] (LinkedIn,  
<<https://www.linkedin.com/in/ly-xiong-bb83ba1b2/>>)

**Maintainer** John Zobitz <zobitz@augsborg.edu>  
**Repository** CRAN  
**Date/Publication** 2024-05-25 17:10:02 UTC

**Contents**

|                                     |           |
|-------------------------------------|-----------|
| acquire_neon_data . . . . .         | 2         |
| check_qf_flags . . . . .            | 3         |
| co2_to_umol . . . . .               | 4         |
| compute_monthly_mean . . . . .      | 5         |
| compute_neon_flux . . . . .         | 6         |
| compute_surface_flux . . . . .      | 7         |
| correct_env_data . . . . .          | 8         |
| dejong_shappert_flux . . . . .      | 8         |
| depth_interpolate . . . . .         | 9         |
| determine_position . . . . .        | 10        |
| diffusivity . . . . .               | 11        |
| env_fingerprint_plot . . . . .      | 12        |
| fit_function . . . . .              | 12        |
| flux_fingerprint_plot . . . . .     | 13        |
| hirano_flux . . . . .               | 14        |
| insert_mean . . . . .               | 15        |
| inside_interval . . . . .           | 15        |
| measurement_detect . . . . .        | 16        |
| measurement_merge . . . . .         | 17        |
| quadrature_error . . . . .          | 18        |
| sjer_env_data_2022_06 . . . . .     | 19        |
| sjer_flux_2022_06 . . . . .         | 19        |
| sjer_megapit_data_2022_06 . . . . . | 20        |
| swc_correct . . . . .               | 21        |
| swc_corrections . . . . .           | 22        |
| tang_2003_flux . . . . .            | 22        |
| tang_2005_flux . . . . .            | 23        |
| <b>Index</b>                        | <b>25</b> |

---

|                   |                                         |
|-------------------|-----------------------------------------|
| acquire_neon_data | <i>Acquire NEON data for processing</i> |
|-------------------|-----------------------------------------|

---

**Description**

Given a site code and dates, apply the neonUtilities package to download the data from NEON API

**Usage**

```
acquire_neon_data(
  site_name,
  download_date,
  time_frequency = "30_minute",
  provisional = FALSE
)
```

**Arguments**

|                |                                                                                                                                                                                                               |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| site_name      | Required. NEON code for a particular site (a string)                                                                                                                                                          |
| download_date  | Required. Date where we end getting NEON data. Format: YYYY-MM (can't specify day). So "2020-05" means it will grab data for the entire 5th month of 2020. (a string). Downloads data for a given month only  |
| time_frequency | Required. Will you be using 30 minute ("30_minute") or 1 minute ("1_minute") recorded data? Defaults to 30 minutes.                                                                                           |
| provisional    | Required. Should you use provisional data when downloading? Defaults to FALSE. See <a href="#">NEON Data Releases</a> . Defaults to FALSE (similar to include.provisional in <a href="#">loadByProduct</a> ). |

**Value**

A list containing stacked environmental data ('site\_data') and soil properties ('site\_megapit').

**Author(s)**

John Zobitz <zobitz@augsborg.edu>

**Examples**

```
out_env_data <- acquire_neon_data("SJER", "2022-06")
```

---

|                |                                                                                          |
|----------------|------------------------------------------------------------------------------------------|
| check_qf_flags | <i>Internal helper function to determine availability of data within a time interval</i> |
|----------------|------------------------------------------------------------------------------------------|

---

**Description**

Given a data measurement, determine when a monthly mean is used and the bulk QF flags. Helps to determine when the environmental measurements produced a QF value and to be used in subsequent flux calculations.

**Usage**

```
check_qf_flags(measurement_name, data)
```

**Arguments**

|                  |                                                                             |
|------------------|-----------------------------------------------------------------------------|
| measurement_name | The name of the measurement (staPres, soilTemp, VSWC, soilCO2Concentration) |
| data             | Data used to check the qf flags                                             |

**Value**

A data frame of startDateTime, horizontalPosition, and the associated QF flag.  
 # changelog and author contributions / copyrights

**Author(s)**

John Zobitz <zobitz@augzburg.edu>

---

|             |                                                                                      |
|-------------|--------------------------------------------------------------------------------------|
| co2_to_umol | <i>Convert co2 concentration from ppm to <math>\mu\text{mol m}^{-3}</math> units</i> |
|-------------|--------------------------------------------------------------------------------------|

---

**Description**

Given a measurement of co2, convert it from ppm to  $\mu\text{mol m}^{-3}$  based on temperature and pressure.  
 Also compute associated error via quadrature.

**Usage**

```
co2_to_umol(
  temperature,
  pressure,
  co2,
  temperature_err,
  pressure_err,
  co2_err,
  zoffset
)
```

**Arguments**

|                 |                                                               |
|-----------------|---------------------------------------------------------------|
| temperature     | Required. Soil temperature (degrees C)                        |
| pressure        | Required. Barometric air pressure (kilopascal)                |
| co2             | Carbon dioxide in ppm                                         |
| temperature_err | Required. Reported Soil temperature error (degrees C)         |
| pressure_err    | Required. Reported Barometric air pressure error (kilopascal) |
| co2_err         | Required. Carbon dioxide in ppm error                         |
| zoffset         | Required. Surface depth (m). Reported as a negative number.   |

**Value**

A value of the converted co2

**Author(s)**

John Zobitz <zobitz@augzburg.edu>

**Examples**

```
co2_to_umol(31,96.3,654,.15,.05,9,-.05)
```

---

compute\_monthly\_mean    *Function to compute monthly means for a given month of NEON data.*

---

**Description**

Given a NEON measurement data frame calculate the monthly mean values across all horizontal and vertical locations. Based off code from Zoey Werbin.

**Usage**

```
compute_monthly_mean(  
  NEON_data,  
  position_columns = c("horizontalPosition", "verticalPosition")  
)
```

**Arguments**

|                  |                                                                                                                                                                  |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NEON_data        | Required. Input vector of neon measurements for a month                                                                                                          |
| position_columns | Optional. Do we group by horizontalPosition, verticalPosition, and? Default is both. Added this option in case we just want to average across a given dimension. |

**Value**

A data frame that reports for each horizontal and vertical position the computed mean and standard deviation from sampling (similar to a bootstrap method) as well as the sample mean and sample standard deviation

**Author(s)**

John Zobitz <zobitz@augzburg.edu>

**References**

Zoey Werbin (@zoey-rw): original author <https://github.com/zoey-rw/microbialForecasts/blob/caa7b1a8aa8a131a5ff9340f15>

## Examples

```
# Download the NEON data directly - here this would be soil moisture
NEON_moist_30m_orig <- neonUtilities::loadByProduct(
  dpID = "DP1.00094.001",
  site = "WREF",
  startdate = "2022-06",
  enddate = "2022-06",
  timeIndex = "30",
  package = "expanded",
  check.size = FALSE,
  include.provisional = TRUE
)

# Then correct the swc
site_swc <- swc_correct(NEON_moist_30m_orig, "WREF", "2022-06")

# Select the columns and the time frequency
time_frequency <- "30_minute"
column_selectors <- c("Mean", "Minimum", "Maximum", "ExpUncert", "StdErMean")

  swc <- site_swc |>
  purrr::pluck(paste0("SWS_", time_frequency)) |>
  dplyr::select(tidyselect::all_of(c("domainID", "siteID",
    "horizontalPosition", "verticalPosition", "startDateTime", "VSWCFinalQF")),
    tidyselect::matches(stringr::str_c("VSWC", column_selectors)))

# Determine a data frame of the different horizontal and vertical positions
swc_positions <- site_swc |>
  purrr::pluck(paste0("sensor_positions_", "00094"))

# Add on the positions for swc
swc <- determine_position(swc_positions, swc)
```

---

|                   |                                      |
|-------------------|--------------------------------------|
| compute_neon_flux | <i>Compute NEON fluxes at a site</i> |
|-------------------|--------------------------------------|

---

## Description

Given a site filename (from `acquire_neon_data`), process and compute fluxes. This file takes a saved data file from `acquire`: 1) Takes the needed components (QF and measurement flags) for soil water, temperature, co2, binding them together in a tidy data frame 2) Interpolates across the measurements 3) Merges air pressure data into this data frame 4) Does a final QF check so we should have only timeperiods where all measurements exist 5) Adds in the megapit data so we have bulk density, porosity measurements at the interpolated depth. 6) Saves the data

## Usage

```
compute_neon_flux(input_site_env, input_site_megapit)
```

**Arguments**

input\_site\_env Required. Input list of environmental data. Usually given from acquire\_neon\_data  
input\_site\_megapit Required. Input list of environmental soil data. Usually given from acquire\_neon\_data

**Value**

Data frame of fluxes and gradient from the timeperiod

**Author(s)**

John Zobitz <zobitz@augsborg.edu> based on code developed by Edward Ayres <eayres@battelleecology.org>

**Examples**

```
# First acquire the NEON data at a given NEON site
out_env_data <- acquire_neon_data("SJER", "2020-05")

# Then process and compute the fluxes:
out_flux <- compute_neon_flux(input_site_env = sjer_env_data_2022_06,
input_site_megapit = sjer_megapit_data_2022_06)
```

---

compute\_surface\_flux    *Internal fucntion to compute CO2 surface flux*

---

**Description**

Computation function. Given a measurement of the co2 and diffusive flux at different levels, return the surface flux

**Usage**

```
compute_surface_flux(input_data)
```

**Arguments**

input\_data Required. A data frame containing zOffsets, diffusivity, and co2 (umol mol-1) and their associated errors

**Value**

A value of the surface CO2 flux (umol m-2 s-1)

**Author(s)**

John Zobitz <zobitz@augsborg.edu>

---

|                  |                                                                                 |
|------------------|---------------------------------------------------------------------------------|
| correct_env_data | <i>Internal function that prepares downloaded NEON data for flux processing</i> |
|------------------|---------------------------------------------------------------------------------|

---

### Description

This file takes data frame from acquire\_neon\_data and: 1) Takes the needed components (QF and measurement flags) for soil water, temperature, co2, binding them together in a tidy data frame 2) Interpolates across the measurements 3) Merges air pressure data into this data frame

### Usage

```
correct_env_data(input_data)
```

### Arguments

|            |                                                     |
|------------|-----------------------------------------------------|
| input_data | Required. Nested data frame from acquire_neon_data. |
|------------|-----------------------------------------------------|

### Value

List of all QF flags over time period and Data frame of environmental measurements for flux computation

### Author(s)

John Zobitz <zobitz@augsborg.edu>

### Examples

```
# Note: you may need to first acquire the NEON data using acquire_neon_data
# Now correct existing environmental data:
corrected_data <- correct_env_data(sjer_env_data_2022_06)
```

---

|                      |                                                                                                            |
|----------------------|------------------------------------------------------------------------------------------------------------|
| dejong_shappert_flux | <i>Internal function to compute surface co2 flux at a given timepoint via De Jong and Schappert (1972)</i> |
|----------------------|------------------------------------------------------------------------------------------------------------|

---

### Description

Given zOffsets, diffusivity, and co2 (umol mol<sup>-1</sup>) and their associated errors, compute the surface flux. This is done by estimating the surface concentration (through linear regression) and doing a gradient calculation. Modified from De Jong and Schappert (1972).

### Usage

```
dejong_shappert_flux(zOffset, co2, co2_err, diffusive, diffusive_err)
```



**Arguments**

|               |                                                                                                 |
|---------------|-------------------------------------------------------------------------------------------------|
| zoffset       | Required. depths below surface - assumed to be positive in value. Important for directionality! |
| co2           | Required. co2 at depth (umol m <sup>-1</sup> )                                                  |
| co2_err       | Required. Associated errors with that value of co2                                              |
| diffusive     | Required. diffusivity at each depth                                                             |
| diffusive_err | Required Associated errors with diffusivity                                                     |

**Value**

Data frame of fluxes associated error

**Author(s)**

John Zobitz <zobitz@augsborg.edu>

**References**

Jong, E. De, and H. J. V. Schappert. 1972. "Calculation of Soil Respiration and Activity from CO<sub>2</sub> Profiles in the Soil." Soil Science 113 (5): 328.

Maier, M., and H. Schack-Kirchner. 2014. "Using the Gradient Method to Determine Soil Gas Flux: A Review." Agricultural and Forest Meteorology 192–193 (July):78–95. <https://doi.org/10.1016/j.agrformet.2014.03.006>.

**See Also**

[hirano\_flux()], [tang\_2003\_flux()], [tang\_2005\_flux()] for other ways to compute surface fluxes.

---

|                   |                                                                      |
|-------------------|----------------------------------------------------------------------|
| depth_interpolate | <i>Internal function to interpolate different depth measurements</i> |
|-------------------|----------------------------------------------------------------------|

---

**Description**

Definition function. Linearly interpolate a measurement across the different measurement depths

**Usage**

```
depth_interpolate(
  input_measurements,
  measurement_name,
  measurement_interpolate
)
```

**Arguments**

- input\_measurements  
Required. Nested data frame (tibble of a months worth of data of co2, temperature, swc, pressure)
- measurement\_name  
Required. Names of measurements we are interpolating. Currently only does one column at a time.
- measurement\_interpolate  
Required. Names of measurement whose depth is used to interpolate (typically co2)

**Value**

A nested data frame with interpolated measurements.

**Author(s)**

John Zobitz <zobitz@augsborg.edu> based on code developed by Edward Ayres <eayres@battelleecology.org>

---

|                    |                                                                  |
|--------------------|------------------------------------------------------------------|
| determine_position | <i>Internal function to determine the depth of a measurement</i> |
|--------------------|------------------------------------------------------------------|

---

**Description**

Given a NEON measurement data frame and measurement depth, pull out the measurement depth for a measurement - because it may vary in time based on if a sensor is replaced.

**Usage**

```
determine_position(input_positions, input_measurement)
```

**Arguments**

- input\_positions  
Required. Input vector of measurements depths
- input\_measurement  
Required. Input vector of measurements.

**Value**

A data frame that reports the measurement depth for and associated environmental measurement.

**Author(s)**

John Zobitz <zobitz@augsborg.edu>

---

|             |                                 |
|-------------|---------------------------------|
| diffusivity | <i>Compute soil diffusivity</i> |
|-------------|---------------------------------|

---

**Description**

Given a tidied data frame of soil measurements (from interpolate.R), compute the diffusivity in a given soil layer

**Usage**

```
diffusivity(
  temperature,
  soil_water,
  pressure,
  temperature_err,
  soil_water_err,
  pressure_err,
  zOffset,
  porVol2To20
)
```

**Arguments**

|                 |                                                                                                                |
|-----------------|----------------------------------------------------------------------------------------------------------------|
| temperature     | Required. Soil temperature (degrees C)                                                                         |
| soil_water      | Required. Soil water content                                                                                   |
| pressure        | Required. Barometric air pressure (kilopascal)                                                                 |
| temperature_err | Required. Reported Soil temperature error (degrees C)                                                          |
| soil_water_err  | Required. Reported Soil water content error                                                                    |
| pressure_err    | Required. Reported Barometric air pressure error (kilopascal)                                                  |
| zOffset         | Required. Measurement level in cm.                                                                             |
| porVol2To20     | Required. Porosity of the 0-20 mm fraction (cm <sup>3</sup> cm <sup>-3</sup> ). Assumes no pores within rocks. |

**Value**

A value of the computed diffusivity

**Author(s)**

John Zobitz <zobitz@augsborg.edu>

**Examples**

```
diffusivity(31,0.0102,96.3,.15,.2135,.05,-.05,0.45)
```

---

|                      |                                                                           |
|----------------------|---------------------------------------------------------------------------|
| env_fingerprint_plot | <i>Helper function to plot QF results for environmental measurements.</i> |
|----------------------|---------------------------------------------------------------------------|

---

**Description**

Given a flux measurement data frame, show when the environmental measurements produced a QF value

**Usage**

```
env_fingerprint_plot(input_fluxes)
```

**Arguments**

input\_fluxes      data frame of computed fluxes

**Value**

A ggplot graph where we have ordered factors showing the QA values a given environmental measurement

**Author(s)**

John Zobitz <zobitz@augzburg.edu>

**Examples**

```
# Make a fingerprint plot for environmental variables:
env_fingerprint_plot(sjer_flux_2022_06)
```

---

|              |                                                                                   |
|--------------|-----------------------------------------------------------------------------------|
| fit_function | <i>Internal function that interpolates a soil measurement to different depths</i> |
|--------------|-----------------------------------------------------------------------------------|

---

**Description**

Definition function. Linearly interpolate a measurement across the different measurement depths

**Usage**

```
fit_function(
  input_depth,
  input_value,
  input_value_err,
  input_value_qf,
  interp_depth,
  measurement_special
)
```

**Arguments**

|                     |                                                                                                                                       |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| input_depth         | Required. Vector of measurement depths.                                                                                               |
| input_value         | Required. Vector of measured values (i.e. soil temperature and soil water) measured at depths input_depth                             |
| input_value_err     | Required. Vector or reported measurement errors. Used to compute prediction error (via quadrature) when linear interpolation is used. |
| input_value_qf      | Required. Vector of qf values from the smoothing 0 = no smoothing, 1 = mean value used (smoothing), 2 = NA value                      |
| interp_depth        | Depths of the sensors required for interpolation                                                                                      |
| measurement_special | Flag if we want to just do linear interpolation for a given measurement                                                               |

**Value**

A data frame of the depth and the measured column for the measurements and reported error

**Author(s)**

John Zobitz <zobitz@augsborg.edu> based on code developed by Edward Ayres <eayres@battelleecology.org>

---

flux\_fingerprint\_plot *Helper function to plot QF results for fluxes.*

---

**Description**

Given a flux measurement data frame, show when the flux and diffusivity measurements produced a QF value

**Usage**

```
flux_fingerprint_plot(input_fluxes)
```

**Arguments**

|              |                               |
|--------------|-------------------------------|
| input_fluxes | data frame of computed fluxes |
|--------------|-------------------------------|

**Value**

A ggplot graph where we have ordered factors showing the QA values a given flux computation

**Author(s)**

John Zobitz <zobitz@augsborg.edu>

## Examples

```
# Make a fingerprint plot for computed flux values:
flux_fingerprint_plot(sjer_flux_2022_06)
```

---

|             |                                                                                                 |
|-------------|-------------------------------------------------------------------------------------------------|
| hirano_flux | <i>Internal function to compute surface co2 flux at a given timepoint via Hirano et al 2005</i> |
|-------------|-------------------------------------------------------------------------------------------------|

---

## Description

Given zOffsets, diffusivity, and co2 (umol mol<sup>-1</sup>) and their associated errors, compute the surface flux. This is done by estimating the surface concentration through linear regression and linear extrapolation of the bottom and top surface fluxes. Modified from Hirano et al (2005).

## Usage

```
hirano_flux(zOffset, co2, co2_err, diffusive, diffusive_err)
```

## Arguments

|               |                                                                                                 |
|---------------|-------------------------------------------------------------------------------------------------|
| zOffset       | Required. depths below surface - assumed to be positive in value. Important for directionality! |
| co2           | Required. co2 at depth (umol m <sup>-1</sup> )                                                  |
| co2_err       | Required. Associated errors with that value of co2                                              |
| diffusive     | Required. diffusivity at each depth                                                             |
| diffusive_err | Required Associated errors with diffusivity                                                     |

## Value

Data frame of fluxes associated error

## Author(s)

John Zobitz <zobitz@augsborg.edu>

## References

Hirano, Takashi, Honghyun Kim, and Yumiko Tanaka. 2003. "Long-Term Half-Hourly Measurement of Soil CO<sub>2</sub> Concentration and Soil Respiration in a Temperate Deciduous Forest." *Journal of Geophysical Research: Atmospheres* 108 (D20). <https://doi.org/10.1029/2003JD003766>.

Maier, M., and H. Schack-Kirchner. 2014. "Using the Gradient Method to Determine Soil Gas Flux: A Review." *Agricultural and Forest Meteorology* 192–193 (July):78–95. <https://doi.org/10.1016/j.agrformet.2014.03.006>.

## See Also

[dejong\_shappert\_flux()], [tang\_2003\_flux()], [tang\_2005\_flux()] for other ways to compute surface fluxes.

---

|             |                                                                                      |
|-------------|--------------------------------------------------------------------------------------|
| insert_mean | <i>Internal function that inserts smoothed mean value of a measurement at a site</i> |
|-------------|--------------------------------------------------------------------------------------|

---

**Description**

Given a site measurement and monthly mean, insert in the monthly mean value when the QF flag fails.

**Usage**

```
insert_mean(data, monthly_mean, measurement_name)
```

**Arguments**

|                  |                                             |
|------------------|---------------------------------------------|
| data             | Required. input data to use                 |
| monthly_mean     | Required. monthly mean of input data to use |
| measurement_name | Required. name of measurement               |

**Value**

Nested data frame of measurements

**Author(s)**

John Zobitz <zobitz@augsborg.edu>

---

|                 |                                                           |
|-----------------|-----------------------------------------------------------|
| inside_interval | <i>Determine if a YYYY-MM string is inside a interval</i> |
|-----------------|-----------------------------------------------------------|

---

**Description**

Determine if a YYYY-MM string is inside a interval

**Usage**

```
inside_interval(start, end, reference_time)
```

**Arguments**

|                |                                           |
|----------------|-------------------------------------------|
| start          | starting interval time                    |
| end            | ending interval time                      |
| reference_time | time we are comparing to - YYYY-MM string |

**Value**

Logical indicating whether or not the reference time is inside the interval. We need this when working with downloaded NEON data which usually comes in a YYYY-MM string

**Author(s)**

John Zobitz <zobitz@augsborg.edu>

**Examples**

```
# Define starting and ending dates:
start_date <- as.POSIXct("2021-06-01 09:30:00",tz="UTC")
end_date <- as.POSIXct("2023-06-01 20:00:00",tz="UTC")

# Test, should return TRUE
inside_interval(start_date,end_date,"2022-06")

# Test, should return FALSE
inside_interval(start_date,end_date,"2020-06")
```

---

|                    |                                                                                                                                                         |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| measurement_detect | <i>Internal function that makes sure for each time, position, and depth we have at least two data points for soil temp and soil h20, 3 for soil co2</i> |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|

---

**Description**

Given a merged data frame of co2, water, and temperature: 1) Filters on QF measurement flags 2) Filters if we have at least 2 soil h20 and temperature measurements, 3 co2 measurements at each time, horizontal position, and vertical depth 3) Filters if we have at least 3 distinct measurements at each time and horizontal position 4) Returns the resulting data frame.

This internal function is created to speed up data processing.

**Usage**

```
measurement_detect(input_data)
```

**Arguments**

|            |                                                                                                                             |
|------------|-----------------------------------------------------------------------------------------------------------------------------|
| input_data | Required. Nested data frame of merged soil water, temperature, co2, and pressure across different NEON locations and depths |
|------------|-----------------------------------------------------------------------------------------------------------------------------|

**Value**

Data frame of fluxes from the timeperiod

**Author(s)**

John Zobitz <zobitz@augsborg.edu> based on code developed by Edward Ayres <eayres@battelleecology.org>



---

|                   |                                                                                           |
|-------------------|-------------------------------------------------------------------------------------------|
| measurement_merge | <i>Internal function that filters environmental data for easier processing of fluxes.</i> |
|-------------------|-------------------------------------------------------------------------------------------|

---

## Description

Internal function. Given a set of environmental measurements, create a merged data frame of measurements and positions where the QF\_flag exists (either observed or a smoothed mean measurement).

## Usage

```
measurement_merge(  
  neon_data,  
  data_code,  
  data_product_id,  
  measurement_name,  
  qf_name  
)
```

## Arguments

|                  |                                                                                                                                           |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| neon_data        | Required. A list of NEON data downloaded from the utilities                                                                               |
| data_code        | Required. Names of data product we are interpolating. (SWS = soil water, ST = soil temperature, SCO2C = soil CO2)                         |
| data_product_id  | Name of the data product 00094 = soil water, 00041 = soil temperature, 00095 = soil CO2                                                   |
| measurement_name | Required. Names of column we are grabbing. (VSWCMean = soil water, soil-TempMean = soil temperature, soilCO2concentrationMean = soil CO2) |
| qf_name          | Required. Names of qf column we are grabbing. (VSWCFinalQF = soil water, finalQF = soil temperature, finalQF = soil CO2)                  |

## Value

A data frame of the requested data.

## Author(s)

John Zobitz <zobitz@augsborg.edu>

quadrature\_error

*Helper function to quickly compute the quadrature error***Description**

Assume a derived quantity  $y$  is a function of inputs  $x_i$ :  $y = f(x_1, x_2, x_3, \dots)$

Given uncertainties ( $x_{\text{err}}$ ) for each  $x_i$ , then this function will compute the corresponding  $y_{\text{err}}$  via quadrature. Inputs are the vector of partial derivatives  $df/dx_i$ , evaluated at  $(x_1, x_2, x_3, \dots)$ .

Resulting  $y_{\text{err}}$  is the square root of the sum of  $(df/dx_1)^2 * (x_{\text{err}})^2 + (df/dx_2)^2 * (x_{\text{err}})^2 + (df/dx_3)^2 * (x_{\text{err}})^2 \dots$

**Usage**

```
quadrature_error(x_pd, x_err)
```

**Arguments**

|                    |                                                                                   |
|--------------------|-----------------------------------------------------------------------------------|
| <code>x_pd</code>  | Required. Input vector of partial derivatives for $y = f(x)$ , evaluated at $x_i$ |
| <code>x_err</code> | Required. Error vector of measurements                                            |

**Value**

A value of quadrature error

**Author(s)**

John Zobitz <zobitz@augsborg.edu>

**Examples**

```
# Let's say we have 5 temperature measurements w/ error::
temperature <- c(31.108, 30.689, 30.463, 30.381, 30.250)
temperature_error <- c(0.1508, 0.1507, 0.1497, 0.1496, 0.1497)

# The sample mean is the sum of all measurements divided by the average:
sum(temperature)/5 # (Can also be computed with mean(temperature))

# The vector of partial derivatives is just 1/n for each measurement:
temperature_pd <- c(1/5, 1/5, 1/5, 1/5, 1/5)
quadrature_error(temperature_pd, temperature_error)
# Note: quadrature_error(1/5, temperature_error) is also allowed.
```

---

sjer\_env\_data\_2022\_06 *Measured environmental data at a NEON site*

---

### Description

A nested dataset containing environmental variables at the SJER site from June 2022. A convenience data frame to make processing fluxes easier for testing. Computed with the function `acquire_neon_data`. All the variables can be used to compute fluxes (along with megapit data).

### Usage

```
data(sjer_env_data_2022_06)
```

### Format

A nested 3 item list

### Details

- `measurement`: Name of environmental measurement
- `data`: Nested list of data corresponding to a measurement
- `monthly_mean`: Monthly mean of measurement, computed in `'compute_monthly_mean'`

---

sjer\_flux\_2022\_06 *Computed flux values at a NEON site*

---

### Description

A nested dataset containing computed fluxes at the SJER site from June 2022. Fluxes were computed with using `compute_neon_flux(sjer_env_data_2022_06,sjer_megapit_data_2022_06)`.

### Usage

```
data(sjer_flux_2022_06)
```

### Format

A nested data frame item list with 7200 rows and 8 columns

**Details**

- `startDateTime`: Time period of measurement (as POSIXct)
- `horizontalPosition`: Sensor location where flux is computed
- `flux_compute`: A nested tibble with variables (1) `flux`, `flux_err`, and `method` (one of 4 implemented)
- `diffusivity`: Computation of surface diffusivity
- `VSWCMeanQF`: QF flag for soil water content across all vertical depths at the given horizontal position: 0 = no issues, 1 = monthly mean used in measurement, 2 = QF fail
- `soilTempMeanQF`: QF flag for soil temperature across all vertical depths at the given horizontal position: 0 = no issues, 1 = monthly mean used in measurement, 2 = QF fail
- `soilCO2concentrationMeanQF`: QF flag for soil CO2 concentration across all vertical depths at the given horizontal position: 0 = no issues, 1 = monthly mean used in measurement, 2 = QF fail
- `staPresMeanQF`: QF flag for atmospheric pressure across all vertical depths at the given horizontal position: 0 = no issues, 1 = monthly mean used in measurement, 2 = QF fail

---

sjer\_megapit\_data\_2022\_06

*Measured soil physical properties at a NEON site*

---

**Description**

A nested dataset containing soil data properties at the SJER site from June 2022. A convenience data frame to make processing fluxes easier for testing. Computed with using the function `acquire_neon_data`. Essentially what is returned when data product DP1.00096.001 is loaded from [loadByProduct](#).

**Usage**

```
data(sjer_megapit_data_2022_06)
```

**Format**

A nested 10 item list

**Details**

- `citation_00096_RELEASE-2024`: BibTex reference
- `issueLog_00096`: Listing of known issues
- `mgp_perarchivesample`: Archived sample information
- `mgp_perbiogeosample`: Data collected on biogeochemistry sample
- `mgp_perbulksample`: Data collected on bulk density sample
- `mgp_perhorizon`: Per soil horizon metadata

- mgp\_permegapit: Data collected per megapit
- readme\_00096: Data product description, issue log, and other metadata about the data product
- validation\_00096: Description of data validation applied at the points of collection and ingest
- variables\_00096: Description and units for each column of data in data tables

## Source

From <https://data.neonscience.org/data-products/DP1.00096.001>

---

|             |                                                                |
|-------------|----------------------------------------------------------------|
| swc_correct | <i>Internal function to correct depths for VSWC NEON data.</i> |
|-------------|----------------------------------------------------------------|

---

## Description

Given the expanded SWC data, return a corrected version based on the values below

## Usage

```
swc_correct(input_swc, curr_site, reference_time)
```

## Arguments

|                |                                                                                         |
|----------------|-----------------------------------------------------------------------------------------|
| input_swc      | Required. input soil water content data from <code>acquire_neon_data</code> (as a list) |
| curr_site      | Current site we are working with                                                        |
| reference_time | Current month we are working with                                                       |

## Value

A revised list of corrected soil water content and depths.

## Author(s)

John Zobitz <zobitz@augsborg.edu>

## Examples

```
# Download the soil water content data:
site_swc <- neonUtilities::loadByProduct(
  dpID="DP1.00094.001",
  site="SJER",
  startdate="2020-05",
  enddate="2020-05",
  timeIndex = "30",
  package="expanded",
  check.size = FALSE,
  include.provisional = TRUE
)
```

```
# Then correct the swc:
site_swc <- swc_correct(site_swc, "SJER", "2020-05")
```

---

|                 |                                                                    |
|-----------------|--------------------------------------------------------------------|
| swc_corrections | <i>Corrected sensor locations for NEON soil water content data</i> |
|-----------------|--------------------------------------------------------------------|

---

### Description

A dataset containing corrected NEON sensor location depths for soil water content data

### Usage

```
swc_corrections
```

### Format

A data frame with 2161 rows and 6 variables

### Details

- domainID (ecological domain of site)
- siteID NEON code to refer to site
- HOR.VER 3 digit code to refer to the horizontal and vertical position of measuring location.
- sensorDepth depth below surface (m)
- startDateTime time measurement was started
- endDateTime time measurement ended

### Source

<<https://data.neonscience.org/data-products/DP1.00094.001>>

---

|                |                                                                                               |
|----------------|-----------------------------------------------------------------------------------------------|
| tang_2003_flux | <i>Internal function to compute surface co2 flux at a given timepoint via Tang et al 2003</i> |
|----------------|-----------------------------------------------------------------------------------------------|

---

### Description

Given zOffsets, diffusivity, and co2 (umol mol<sup>-1</sup>) and their associated errors, compute the surface flux. This is done by computing the gradient of co2 (slope from linear regression) and linear extrapolation of surface diffusivity. Modified from Tang et al (2003).

**Usage**

```
tang_2003_flux(zOffset, co2, co2_err, diffusive, diffusive_err)
```

**Arguments**

|               |                                                                                                 |
|---------------|-------------------------------------------------------------------------------------------------|
| zOffset       | Required. depths below surface - assumed to be positive in value. Important for directionality! |
| co2           | Required. co2 at depth (umol m <sup>-1</sup> )                                                  |
| co2_err       | Required. Associated errors with that value of co2                                              |
| diffusive     | Required. diffusivity at each depth                                                             |
| diffusive_err | Required Associated errors with diffusivity                                                     |

**Value**

Data frame of fluxes associated error

**Author(s)**

John Zobitz <zobitz@augsborg.edu>

**References**

Tang, Jianwu, Dennis D Baldocchi, Ye Qi, and Liukang Xu. 2003. "Assessing Soil CO<sub>2</sub> Efflux Using Continuous Measurements of CO<sub>2</sub> Profiles in Soils with Small Solid-State Sensors." *Agricultural and Forest Meteorology* 118 (3): 207–20. [https://doi.org/10.1016/S0168-1923\(03\)00112-6](https://doi.org/10.1016/S0168-1923(03)00112-6).

Maier, M., and H. Schack-Kirchner. 2014. "Using the Gradient Method to Determine Soil Gas Flux: A Review." *Agricultural and Forest Meteorology* 192–193 (July):78–95. <https://doi.org/10.1016/j.agrformet.2014.03.006>.

**See Also**

[dejong\_shappert\_flux()], [hirano\_flux()], [tang\_2005\_flux()] for other ways to compute surface fluxes.

---

|                |                                                                                               |
|----------------|-----------------------------------------------------------------------------------------------|
| tang_2005_flux | <i>Internal function to compute surface co2 flux at a given timepoint via Tang et al 2005</i> |
|----------------|-----------------------------------------------------------------------------------------------|

---

**Description**

Given zOffsets, diffusivity, and co2 (umol mol<sup>-1</sup>) and their associated errors, compute the surface flux. This is done by linear extrapolation of surface fluxes. Modified from Tang et al (2005).

**Usage**

```
tang_2005_flux(zOffset, co2, co2_err, diffusive, diffusive_err)
```

**Arguments**

|               |                                                                                                 |
|---------------|-------------------------------------------------------------------------------------------------|
| zOffset       | Required. depths below surface - assumed to be positive in value. Important for directionality! |
| co2           | Required. co2 at depth (umol m <sup>-1</sup> )                                                  |
| co2_err       | Required. Associated errors with that value of co2                                              |
| diffusive     | Required. diffusivity at each depth                                                             |
| diffusive_err | Required Associated errors with diffusivity                                                     |

**Value**

Data frame of fluxes associated error

**Author(s)**

John Zobitz <zobitz@augsborg.edu>

**References**

- Tang, Jianwu, Laurent Misson, Alexander Gershenson, Weixin Cheng, and Allen H. Goldstein. 2005. "Continuous Measurements of Soil Respiration with and without Roots in a Ponderosa Pine Plantation in the Sierra Nevada Mountains." *Agricultural and Forest Meteorology* 132 (3): 212–27. <https://doi.org/10.1016/j.agrformet.2005.07.011>.
- Maier, M., and H. Schack-Kirchner. 2014. "Using the Gradient Method to Determine Soil Gas Flux: A Review." *Agricultural and Forest Meteorology* 192–193 (July):78–95. <https://doi.org/10.1016/j.agrformet.2014.03.006>.

**See Also**

[dejong\_shappert\_flux()], [hirano\_flux()], [tang\_2003\_flux()] for other ways to compute surface fluxes.



# Index

- \* **Currently**
  - compute\_monthly\_mean, [5](#)
- \* **datasets**
  - sjer\_env\_data\_2022\_06, [19](#)
  - sjer\_flux\_2022\_06, [19](#)
  - sjer\_megapit\_data\_2022\_06, [20](#)
  - swc\_corrections, [22](#)
- \* **none**
  - compute\_monthly\_mean, [5](#)

[acquire\\_neon\\_data, 2](#)

[check\\_qf\\_flags, 3](#)  
[co2\\_to\\_umol, 4](#)  
[compute\\_monthly\\_mean, 5](#)  
[compute\\_neon\\_flux, 6](#)  
[compute\\_surface\\_flux, 7](#)  
[correct\\_env\\_data, 8](#)

[dejong\\_shappert\\_flux, 8](#)  
[depth\\_interpolate, 9](#)  
[determine\\_position, 10](#)  
[diffusivity, 11](#)

[env\\_fingerprint\\_plot, 12](#)

[fit\\_function, 12](#)  
[flux\\_fingerprint\\_plot, 13](#)

[hirano\\_flux, 14](#)

[insert\\_mean, 15](#)  
[inside\\_interval, 15](#)

[loadByProduct, 3, 20](#)

[measurement\\_detect, 16](#)  
[measurement\\_merge, 17](#)

[quadrature\\_error, 18](#)

[sjer\\_env\\_data\\_2022\\_06, 19](#)  
[sjer\\_flux\\_2022\\_06, 19](#)  
[sjer\\_megapit\\_data\\_2022\\_06, 20](#)  
[swc\\_correct, 21](#)  
[swc\\_corrections, 22](#)  
[tang\\_2003\\_flux, 22](#)  
[tang\\_2005\\_flux, 23](#)