

# Package ‘neonPlantEcology’

July 22, 2025

**Type** Package

**Title** Process NEON Plant Data for Ecological Analysis

**Version** 1.6.1

**Date** 2024-05-30

**Maintainer** Adam Mahood <admahood@gmail.com>

**Description** Downloading and organizing plant presence and percent cover data from the National Ecological Observatory Network <<https://www.neonscience.org>>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Imports** neonUtilities, vegan, ggplot2, data.table, dtplyr, dplyr,  
lubridate, sf, stringr, tibble, tidyr, ggpubr, utils

**RoxygenNote** 7.3.1

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**URL** <https://github.com/admahood/neonPlantEcology>

**BugReports** <https://github.com/admahood/neonPlantEcology/issues>

**NeedsCompilation** no

**Author** Adam Mahood [aut, cre],  
Jacob Macdonald [ctb],  
Ranjan Muthukrishnan [ctb]

**Depends** R (>= 3.5.0)

**Repository** CRAN

**Date/Publication** 2024-05-30 17:30:02 UTC

Contents

D14 . . . . .	2
npe_change_native_status . . . . .	3
npe_cm_metadata . . . . .	3
npe_community_matrix . . . . .	4
npe_download . . . . .	5
npe_eventID_fixer . . . . .	6
npe_groundcover . . . . .	7
npe_heights . . . . .	8
npe_longform . . . . .	9
npe_plot_centroids . . . . .	10
npe_site_ids . . . . .	11
npe_summary . . . . .	11
npe_update_subplots . . . . .	13
plot_centroids . . . . .	13
sites . . . . .	14
site_polygons . . . . .	14
<b>Index</b>	<b>16</b>

---

D14	<i>Plant Presence and Percent Cover Data for Domain 14</i>
-----	--

---

Description

This includes Jornada Experimental Range and Santa Rita Experimental Range

Usage

D14

Format

## 'D14' A list with 8 items, 2 of which are used by neonPlantEcology

Source

<<https://doi.org/10.48443/9579-a253>>  
<<https://data.neonscience.org/data-products/DP1.10058.001>>

---

`npe_change_native_status`*Change the native status code for a particular taxon at a particular site*

---

### Description

Sometimes even though a particular species identity is not known, the end user can still determine its native status. For example, maybe the taxon was identified to the genus level, and the local flora confirms that all plants in that genus are native at that particular site. This function allows for post-hoc modification of the native status code for cases like this.

### Usage

```
npe_change_native_status(df, taxon, site, new_code)
```

### Arguments

<code>df</code>	is the data frame returned by <code>npe_longform</code>
<code>taxon</code>	is the <code>taxonID</code> column in the data frame
<code>site</code>	is the identity of the NEON site (e.g. "JORN")
<code>new_code</code>	is the <code>NativeStatusCode</code> value to change to

### Value

a data frame

### Examples

```
data("D14")
lf_div <- npe_longform(D14)
modified_lf_div <- npe_change_native_status(lf_div, "ABUTI", "JORN", "N")
```

---

`npe_cm_metadata`*Get plot information from a community matrix*

---

### Description

The `npe_community_matrix()` function is designed to work with the `vegan` package, and one of the requirements of `vegan` functions is that there are only numeric columns in community matrices. Therefore, all of the metadata is collapsed into the rownames. This function allows you to extract that very basic metadata back out to a more easily interpretable data frame.

**Usage**

```
npe_cm_metadata(comm)
```

**Arguments**

comm                    the community matrix object created by npe\_community\_matrix()

**Value**

a data frame

**Examples**

```
data("D14")
npe_community_matrix(D14) |> npe_cm_metadata()
```

---

npe\_community\_matrix    *Create a species abundance or occurrence matrix*

---

**Description**

npe\_community\_matrix creates a wide matrix of species cover or binary (presence/absence) values with the plot/subplot/year as rownames. This is useful for the vegan package, hence the name.

**Usage**

```
npe_community_matrix(
  x,
  scale = "plot",
  trace_cover = 0.5,
  timescale = "annual",
  input = "neon_div_object",
  binary = FALSE
)
```

**Arguments**

x	Input object. See input argument help for more details.
scale	what level of aggregation? This can be "1m", "10m", "100m", "plot", which is the default, or "site".
trace_cover	cover value for subplots where only occupancy was recorded
timescale	what temporal resolution? can be "subannual", which is really only applicable at sites where there are multiple bouts per year, "annual" or "all", which dissolves together the entire time series.

input	by default, longform dataframe is calculated from the diversity object and then converted to a community matrix, set this option to "lf" to use a longform data frame that was created separately (and perhaps modified). Another option is input = "divStack", which is using the output from the divStack function in the neonPlants package. Using a premade longform data frame or a divStack output will use the spatial and temporal scale of that input data separately
binary	should the matrix be converted from percent cover to binary?
neon_div_object	the raw diversity data downloaded using neonPlantEcology::download_plant_div() or the function neonUtilities::loadByProduct() with the dpID argument set to "DP1.10058.001".

### Value

a data frame with each row a site aggregated at the spatial and temporal scales chosen by the user. Each column is a single species, and cell values can be either cover (a value between 0 and 100) or occurrence (1 or 0)

### Examples

```
data("D14")
comm <- npe_community_matrix(D14)
```

---

npe\_download

*Data downloader*


---

### Description

A wrapper function to download data from the NEON API using neonUtilities::loadByProduct. Some commonly used products are provided as plain language options, otherwise the user can enter the product ID number (dpID). Downloads Plant Presence and Percent Cover by default (DP1.10058.001).

### Usage

```
npe_download(
  sites = "JORN",
  dpID = NA,
  token = NA,
  verbose = TRUE,
  product = "plant_diversity",
  ...
)
```

**Arguments**

sites	a vector of NEON site abbreviations. Defaults to "JORN"
dpID	if you need a data product not given as one of the product options, set the data product ID here (e.g. "DP1.10023.001").
token	a token from neonscience.org
verbose	if true, prints which sites are being downloaded
product	a plain language vector of the data product to be downloaded. Can be "plant_diversity", "litterfall", "woody_veg_structure", "belowground_biomass", "herbaceous_clip", "coarse_downed_wood", or "soil_microbe_biomass"
...	additional arguments can be passed to neonUtilities::loadByProduct see ?neonUtilities::loadByProduct for more details

**Value**

a list

**Examples**

```
diversity_object <- npe_download(sites = "JORN")
```

---

npe_eventID_fixer	<i>fix errors in the eventID column</i>
-------------------	---

---

**Description**

neonPlantEcology is a house of cards that rests delicately upon the eventID column being in the site.bout-number.year format, and if there is any deviation from that format all hell breaks loose. This function converts any NA or non-standard eventID rows to the desired format.

**Usage**

```
npe_eventID_fixer(x, verbose = FALSE)
```

**Arguments**

x	raw list data from NEON api
verbose	if true, prints details of which eventID errors were fixed into the console

**Value**

the same list object but with repaired eventIDs

**Examples**

```
data("D14")
x <- npe_eventID_fixer(D14)
```

---

npe_groundcover	<i>Get ground cover and other variables</i>
-----------------	---

---

## Description

Get ground cover and other variables

## Usage

```
npe_groundcover(  
  neon_div_object,  
  scale = "plot",  
  verbose = FALSE,  
  pc_na_value = 0.5,  
  timescale = "annual"  
)
```

## Arguments

neon_div_object	the raw diversity data downloaded using <code>neonPlantEcology::download_plant_div()</code> or the function <code>neonUtilities::loadByProduct()</code> with the <code>dpID</code> argument set to "DP1.10058.001".
scale	the spatial scale of aggregation. Can be "1m", "10m", "100m", "plot" or "site". default is "plot".
verbose	if true, prints details of which eventID errors were fixed into the console
pc_na_value	sometimes the raw data from neon will have NA's in the percent cover cells. This is assumed to be a data entry error and is set to 0.5 by default.
timescale	The temporal scale of aggregation. Can be "all", "annual" or "subannual" in the case of multiple sampling bouts per year. Defaults to "annual".

## Value

a data frame with each row a single observation of ground cover at the spatial and temporal scale chosen by the user.

## Examples

```
data("D14")  
groundcover <- npe_groundcover(D14)
```

---

npe\_heights

*Get heights*


---

## Description

Get heights

## Usage

```
npe_heights(
  neon_div_object,
  scale = "plot",
  verbose = FALSE,
  timescale = "annual"
)
```

## Arguments

neon_div_object	the raw diversity data downloaded using <code>neonPlantEcology::download_plant_div()</code> or the function <code>neonUtilities::loadByProduct()</code> with the <code>dpID</code> argument set to "DP1.10058.001".
scale	the spatial scale of aggregation. Can be "1m", "10m", "100m", "plot" or "site". default is "plot".
verbose	if true, prints details of which eventID errors were fixed into the console
timescale	The temporal scale of aggregation. Can be "all", "annual" or "subannual" in the case of multiple sampling bouts per year. Defaults to "annual".

## Value

a data frame with each row a single observation of species height at the spatial and temporal scale chosen by the user.

## Examples

```
data("D14")
heights <- npe_heights(D14)
```



---

npe_longform	<i>Convert raw NEON diversity object to longform plant cover data frame</i>
--------------	---

---

## Description

The diversity data from NEON comes as a list containing 2 data frames of data that need to be combined, among other things. Here, we take those two data frames and combine them into a longform data frame that can then be further modified for analysis. Most of the unnecessary information from the raw data has been removed. Column names that remain are plotID, subplotID, year, taxonID, cover, scientificName, nativeStatusCode, family, and site.

## Usage

```
npe_longform(
  neon_div_object,
  trace_cover = 0.5,
  pc_na_value = 0.5,
  scale = "plot",
  verbose = FALSE,
  timescale = "annual"
)
```

## Arguments

neon_div_object	the raw diversity data downloaded using <code>neonPlantEcology::download_plant_div()</code> or the function <code>neonUtilities::loadByProduct()</code> with the <code>dpID</code> argument set to "DP1.10058.001".
trace_cover	cover value for subplots where only occupancy was recorded
pc_na_value	sometimes the raw data from neon will have NA's in the percent cover cells. This is assumed to be a data entry error and is set to 0.5 by default.
scale	what level of spatial aggregation? This can be "1m", "10m", "100m", "plot", which is the default, or "site".
verbose	if true, prints details of which eventID errors were fixed into the console
timescale	what level of temporal aggregation? can be "subannual", which is only important for sites with multiple sampling bouts per year, "annual" or "all" for the full time series.

## Value

a data frame with each row a single observation of species cover at the spatial and temporal scale chosen by the user.

**Examples**

```
data("D14")
lf <- npe_longform(D14)
```

---

npe_plot_centroids	<i>Download and join spatial information to a neonPlantEcology output data frame</i>
--------------------	--

---

**Description**

Download and join spatial information to a neonPlantEcology output data frame

**Usage**

```
npe_plot_centroids(
  df,
  type = "latlong",
  spatial_only = TRUE,
  input = "community_matrix"
)
```

**Arguments**

df	a neonPlantEcology-produced data frame
type	what type of ancillary data structure you want joined. Can be "spatial", which will turn the data frame into an sf data frame, or "latlong", which will add the latitudes and longitudes and other ancillary data as columns only.
spatial_only	set to TRUE if you only want the coordinates and none of the ancillary variables.
input	to what kind of neonPlantEcology product are you appending? Can be "community_matrix", "longform_cover", or "summary_info".

**Value**

a data frame

---

npe_site_ids	<i>Get site ids</i>
--------------	---------------------

---

### Description

This returns a list of 4 letter site ID codes to feed into `npe_download`. It can return all 47 siteID codes, or a subset based on site type, aridity index, Koppen-Geiger Climate region, or NEON domain.

### Usage

```
npe_site_ids(by, domain = NA, type = NA, aridity = NA, koppen = NA)
```

### Arguments

<code>by</code>	How to select sites? Can be "all", "domain", "ai", "koppen", or "type".
<code>domain</code>	can be one or more domain codes, as a character vector, or as a number. e.g. <code>domain = c("D01", "D14")</code> , or <code>domain = c(3, 14)</code> , can also be a mix: <code>domain = c(3, "D04")</code> .
<code>type</code>	can be "Core Terrestrial" or "Relocatable Terrestrial"
<code>aridity</code>	can be "Hyper-Arid", "Arid", "Dry sub-humid", or "Humid"
<code>koppen</code>	can be any 3 letter Koppen-Geiger code, or one of "Equatorial", "Arid", "Temperate", "Boreal", "Polar"

### Value

a vector of four letter site identification codes.

### Examples

```
all_sites <- npe_site_ids(by = "all")
npe_site_ids(by = "domain", domain = c("Northeast", "Mid-Atlantic"))
npe_site_ids(by = "domain", domain = c("D02", 15))
```

---

npe_summary	<i>Get plant biodiversity information for NEON plots</i>
-------------	--

---

### Description

`npe_summary` calculates various biodiversity and cover indexes at the plot or subplot scale at each timestep for each plot. Outputs a data frame with number of species, percent cover, relative percent cover (relative to the cover of the other plants), and shannon diversity, for natives, exotics, "notexotics", unknowns, and all species. "notexotic" refers to all species with native or unknown origin status. Also calculates all of these metrics for the families of your choice.

**Usage**

```
npe_summary(
  neon_div_object,
  scale = "plot",
  trace_cover = 0.5,
  timescale = "annual",
  betadiversity = FALSE,
  families = NA
)
```

**Arguments**

neon_div_object	the raw <code>vegan::diversity</code> data downloaded using <code>neonPlantEcology::download_plant_div()</code> or #' the function <code>neonUtilities::loadByProduct()</code> with the <code>dpID</code> argument set to "DP1.10058.001".
scale	what level of aggregation? This can be "1m", "10m", "100m", "plot" or "site". "plot" is the default.
trace_cover	cover value for subplots where only occupancy was recorded
timescale	by default <code>npe_summary</code> groups everything by year. The user may set this argument to "all" to have the function aggregate the years together and then calculate diversity and cover indexes, or "subannual" for bout-level.
betadiversity	If evaluating at the plot or site level, should beta diversity (turnover and nestedness) be calculated. If <code>scale = plot</code> , it will calculate betadiversity within each plot, using the combined species presences within the 1 and 10 m subplots, and so it's calculated from 8 subplots before 2020, 6 after. if <code>scale = site</code> , it calculates the betadiversity between plots.
families	Which specific families should the metrics be calculated for? This can be a concatenated vector if the user want more than one family.

**Value**

a data frame of higher-level summary information. Number of species, Shannon-Weiner alpha diversity, cover, relative cover, for all species together and grouped by `nativeStatusCode`.

**Examples**

```
data("D14")
plot_level <- neonPlantEcology::npe_summary(neon_div_object = D14, scale = "plot")
```

---

npe_update_subplots	<i>Change subplot names</i>
---------------------	-----------------------------

---

**Description**

The 2024 release features a change in subplot names. This function changes subplot names of the Plant Presence and Percent Cover raw list object from the old format to the new format, to ensure backwards compatibility. This is mostly an internal helper function

**Usage**

```
npe_update_subplots(neon_div_object)
```

**Arguments**

neon\_div\_object  
a list downloaded using npe\_download containing Plant Presence and Percent Cover data

**Value**

a

**Examples**

```
data("D14")  
D14_updated <- npe_update_subplots(D14)
```

---

plot_centroids	<i>Plot centroids for the entire NEON network</i>
----------------	---

---

**Description**

Plot centroids for the entire NEON network

**Usage**

```
plot_centroids
```

**Format**

## 'plot\_centroids' A simple feature collection with 3842 features and 36 fields

**Source**

<<https://www.neonscience.org>>

---

sites	<i>National Ecological Observatory Network Core and Relocatable Terrestrial Sites</i>
-------	---

---

### Description

Note: Some sites have more than one polygon. There are 59 polygons and 47 total sites.

### Usage

sites

### Format

## 'sites' data frame with 47 features and 8 fields

**domainNumb** Domain Number

**domainName** Domain Name

**siteType** Site type. Core or Relocatable

**siteID** Four letter site ID. Used in npe\_download()

**koppen\_fine** Koppen-Geiger climate classification from Beck et al 2023

**koppen\_coarse** Coarsest category of K-G climate classification from Beck et al 2023

**ai** Annual aridity index from Zomer & Trabucco 2022

**ai\_class** Climate classification based on the aridity index from Zomer & Trabucco 2022

### Source

<<https://www.neonscience.org>>

<<https://doi.org/10.6084/m9.figshare.7504448.v5>>

<<https://doi.org/10.1038/s41597-023-02549-6>>

---

site_polygons	<i>Polygons of National Ecological Observatory Network Core and Relocatable Terrestrial Sites</i>
---------------	---

---

### Description

Note: Some sites have more than one polygon. There are 59 polygons and 47 total sites.

### Usage

site\_polygons

### Format

## 'site\_polygons' A simple feature collection with 59 features and 8 fields

**domainNumb** Domain Number

**domainName** Domain Name

**siteType** Site type. Core or Relocatable

**siteName** Site name

**siteID** Four letter site ID. Used in npe\_download()

**siteHost** Organization hosting the site

**areaKm2** Area of the site in square kilometers

**acres** Area of the site in acres

**geometry** list column containing geometry information for each polygon

### Source

<<https://www.neonscience.org>>

# Index

- \* **datasets**
  - D14, [2](#)
  - plot\_centroids, [13](#)
  - site\_polygons, [14](#)
  - sites, [14](#)
- \* **diversity**
  - npe\_download, [5](#)
- \* **download**
  - npe\_download, [5](#)
- \* **neon**
  - npe\_download, [5](#)

D14, [2](#)

npe\_change\_native\_status, [3](#)

npe\_cm\_metadata, [3](#)

npe\_community\_matrix, [4](#)

npe\_download, [5](#)

npe\_eventID\_fixer, [6](#)

npe\_groundcover, [7](#)

npe\_heights, [8](#)

npe\_longform, [9](#)

npe\_plot\_centroids, [10](#)

npe\_site\_ids, [11](#)

npe\_summary, [11](#)

npe\_update\_subplots, [13](#)

plot\_centroids, [13](#)

site\_polygons, [14](#)

sites, [14](#)