

# Package: ReLTER (via r-universe)

December 4, 2024

**Type** Package

**Title** An Interface for the eLTER Community

**Version** 2.1.1

**Maintainer** Alessandro Oggioni <alessandro.oggioni@cnr.it>

**Description** ReLTER provides access to DEIMS-SDR (<https://deims.org/>), and allows interaction with data and software implemented by eLTER Research Infrastructure (RI) thus improving data sharing among European LTER projects. ReLTER uses the R language to access and interact with the DEIMS-SDR archive of information shared by the Long Term Ecological Research (LTER) network. This package grew within eLTER H2020 as a major project that will help advance the development of European Long-Term Ecosystem Research Infrastructures (eLTER RI - <https://elter-ri.eu>). The ReLTER package functions in particular allow to: - retrieve the information about entities (e.g. sites, datasets, and activities) shared by DEIMS-SDR (see e.g. `get_site_info` function); - interact with the [ODSEurope]([maps.opendatascience.eu](https://maps.opendatascience.eu)) starting with the dataset shared by [DEIMS-SDR](<https://deims.org/>) (see e.g. `get_site_ODS`)([https://docs.ropensci.org/ReLTER/reference/get\\_site\\_ODS.html](https://docs.ropensci.org/ReLTER/reference/get_site_ODS.html)) function); - use the eLTER site informations to download and crop geospatial data from other platforms (see e.g. `get_site_ODS` function); - improve the quality of the dataset (see e.g. `get_id_worms`). Functions currently implemented are derived from discussions of the needs among the eLTER users community. The ReLTER package will continue to follow the progress of eLTER-RI and evolve, adding new tools and improvements as required.

**License** GPL (>= 3)

**URL** <https://github.com/ropensci/ReLTER>

**BugReports** <https://github.com/ropensci/ReLTER/issues>

**Depends** R (>= 3.5.0)

**Imports** countrycode, dplyr, dtplyr, geodata, ggforce, ggplot2, grDevices, grid, gridExtra, httr, httr2, jqr, jsonlite, leaflet (>= 2.1.1), lifecycle, lubridate, magrittr, MODISTsp, purrr, qrcode, raster (>= 3.3-13), RColorBrewer, RCurl, Rdpack, readr, rnaturalearth, rosm, rworldmap, sf (>= 0.9-5), spocc, stringi, stringr, taxize (>= 0.9.97), terra, tibble, tidyr, tmap (>= 3.1), units, utils, waffle, webshot, worrms, XML, xml2, xslt, zen4R

**Suggests** covr, httpptest, ISOcodes, knitr, rmarkdown, rnaturalearthdata, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**RdMacros** Rdpack

**Config/testthat/edition** 3

**Encoding** UTF-8

**Language** en-GB

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.3

**Config/pak/sysreqs** libfontconfig1-dev libfreetype6-dev libfribidi-dev libgdal-dev gdal-bin libgeos-dev libglpk-dev make libharfbuzz-dev libmagick++-dev gsfonts libicu-dev libjpeg-dev libpng-dev libprotobuf-dev libsecret-1-dev libtiff-dev libxml2-dev libssl-dev libproj-dev protobuf-compiler libprotoc-dev libsqlite3-dev libudunits2-dev libnode-dev libx11-dev zlib1g-dev

**Repository** <https://ropensci.r-universe.dev>

**RemoteUrl** <https://github.com/ropensci/ReLTER>

**RemoteRef** main

**RemoteSha** 1715916c18530717f4292cbbbb20025f28060de3

## Contents

elter_write_rdata . . . . .	3
get_activity_info . . . . .	4
get_dataset_info . . . . .	6
get_deims_API_version . . . . .	8
get_deims_base_url . . . . .	8
get_ilter_envcharacters . . . . .	9
get_ilter_generalinfo . . . . .	10
get_ilter_observedProperties . . . . .	11
get_ilter_research_topics . . . . .	12
get_network_envcharacters . . . . .	13
get_network_observedProperties . . . . .	15

get_network_related_resources . . . . .	16
get_network_research_topics . . . . .	17
get_network_sites . . . . .	18
get_sensor_info . . . . .	19
get_sensor_observed_properties . . . . .	21
get_site_info . . . . .	22
get_site_MODIS . . . . .	24
get_site_ODS . . . . .	28
get_site_speciesOccurrences . . . . .	30
get_sos_foi . . . . .	32
get_sos_obs . . . . .	34
get_sos_procedurelist . . . . .	36
get_sos_procedure_info . . . . .	37
get_zenodo_data . . . . .	39
map_occ_gbif2elter . . . . .	40
map_occ_inat2elter . . . . .	41
map_occ_obis2elter . . . . .	42
package_settings . . . . .	43
plot_agg_map . . . . .	43
plot_timeseries . . . . .	44
produce_network_points_map . . . . .	45
produce_site_map . . . . .	47
produce_site_observedProperties_pie . . . . .	51
produce_site_observedProperties_waffle . . . . .	52
produce_site_qrcode . . . . .	54
produce_zenodo_record . . . . .	55
produce_zenodo_record_from_elter_reporting . . . . .	58
reporting_compose_file_name . . . . .	61
reporting_produce_data_object_v1.3 . . . . .	63
reporting_save_archive . . . . .	65
save_occ_eLTER_reporting_Archive . . . . .	67
set_deims_base_url . . . . .	68
taxon_id_pesi . . . . .	68
taxon_id_worms . . . . .	70

**Index** **72**

elter\_write\_rdata      *eLTER write Rdata data*

**Description**

**[Experimental]** This function write a Rdata file from csv, tsv, txt, xls or xlsx dataset

**Usage**

elter\_write\_rdata(myfiles, delim)

**Arguments**

myfiles	A character. The list of the files to deposit in Zenodo. Please provide all files only with 'csv' extension.
delim	A character. Provide the character used to separate fields within a record. Only if the extension of the file(s) are 'csv', 'tsv', or 'txt'.

**Value**

This function returns a rds files.

**Author(s)**

Alessandro Oggioni, PhD <oggioni.a@irea.cnr.it>

**Examples**

```
## Not run:
## Not run:

elter_write_rdata(
  myfiles = c(
    "miscellaneous/file_show/data_mapping.csv",
    "miscellaneous/file_show/reference_TAXA.csv",
    "miscellaneous/file_show/reference_VARIABLES.csv"
  ),
  delim = ";"
)

## End(Not run)

## End (Not run)
```

---

get\_activity\_info      *Obtain the information about of an eLTER activity.*

---

**Description**

[Stable] This function obtains the information about of an eLTER activity (e.g. <https://deims.org/activity/8786fc6d-5d70-495c-b901-42f480182845>) provided in DEIMS-SDR catalogue.

**Usage**

```
get_activity_info(activityid, show_map = FALSE)
```

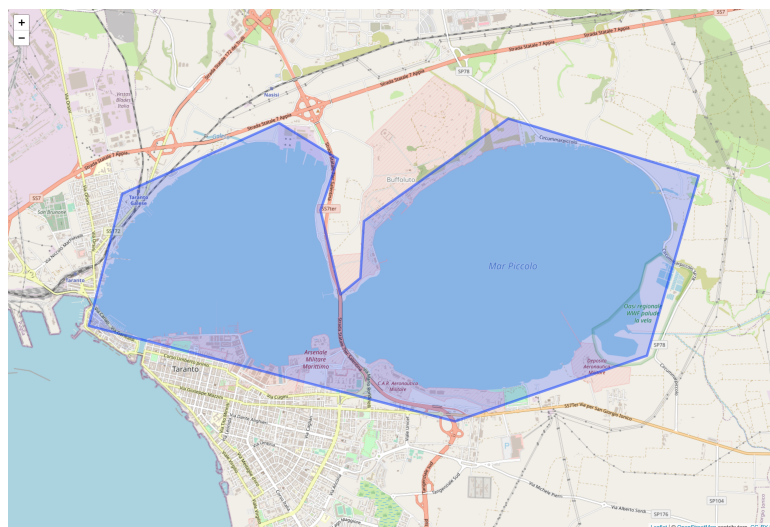
### Arguments

activityid	A character. It is the DEIMS ID of activity make from DEIMS-SDR website. DEIMS ID information <a href="#">here</a> . The DEIMS.id of activity is the URL for the activity page.
show_map	A boolean. If TRUE a Leaflet map with occurrences is shown. Default FALSE.

### Value

The output of the function is a tibble with main features of the activities in a site, and a leaflet map plot.

### The function output



### Author(s)

Alessandro Oggioni, PhD (2020) <oggioni.a@irea.cnr.it>

### References

- Wickham H, François R, Henry L, Müller K (2022). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.9, <https://CRAN.R-project.org/package=dplyr>.
- R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.
- Pebesma E (2018). "Simple Features for R: Standardized Support for Spatial Vector Data." *The R Journal*, **10**(1), 439–446. doi:10.32614/RJ2018009.
- Cheng J, Karambelkar B, Xie Y (2022). *leaflet: Create Interactive Web Maps with the JavaScript 'Leaflet' Library*. R package version 2.1.1, <https://CRAN.R-project.org/package=leaflet>.

## Examples

```
activities <- get_activity_info(  
  activityid =  
    "https://deims.org/activity/8786fc6d-5d70-495c-b901-42f480182845",  
  show_map = FALSE  
)  
activities
```

---

get_dataset_info	<i>Obtain the information about of an eLTER dataset.</i>
------------------	--

---

## Description

**[Stable]** This function obtains the information about of an eLTER dataset (e.g. <https://deims.org/activity/8786fc6d-5d70-495c-b901-42f480182845>) provided in **DEIMS-SDR catalogue**.

## Usage

```
get_dataset_info(datasetid, show_map = FALSE)
```

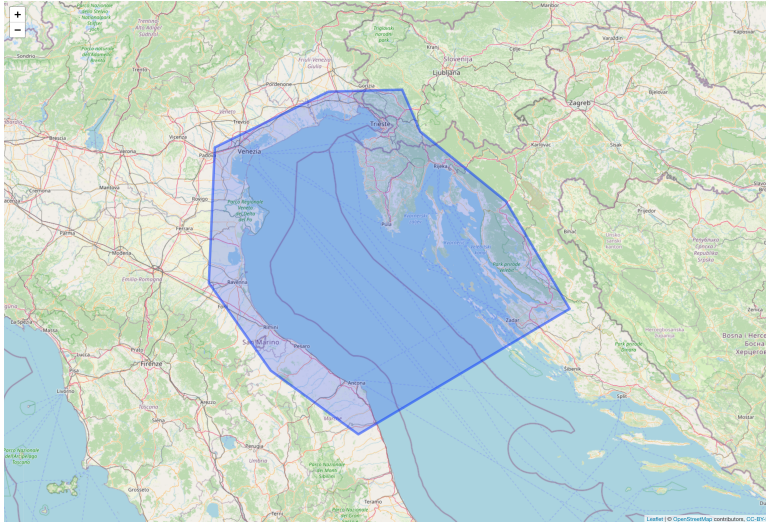
## Arguments

datasetid	A character. It is the DEIMS ID of dataset make from DEIMS-SDR website. DEIMS ID information <a href="#">here</a> . The DEIMS ID of dataset is the URL for the dataset page.
show_map	A boolean. If TRUE a Leaflet map with occurrences is shown. Default FALSE.

## Value

The output of the function is a tibble with main features of the site and the related resources collected by site.

### The function output



### Author(s)

Alessandro Oggioni, PhD (2020) <oggioni.a@irea.cnr.it>

### References

Wickham H, François R, Henry L, Müller K (2022). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.9, <https://CRAN.R-project.org/package=dplyr>.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.

Pebesma E (2018). “Simple Features for R: Standardized Support for Spatial Vector Data.” *The R Journal*, **10**(1), 439–446. doi:10.32614/RJ2018009.

Cheng J, Karambelkar B, Xie Y (2022). *leaflet: Create Interactive Web Maps with the JavaScript 'Leaflet' Library*. R package version 2.1.1, <https://CRAN.R-project.org/package=leaflet>.

### Examples

```
tDataset <- get_dataset_info(  
  datasetid =  
    "https://deims.org/dataset/38d604ef-decb-4d67-8ac3-cc843d10d3ef",  
  show_map = TRUE  
)  
tDataset
```

---

get\_deims\_API\_version *Get version of DEIMS-SDR API*

---

**Description**

**[Stable]** This function obtains the version of the DEIMS-SDR API.

**Usage**

```
get_deims_API_version(deims_url = get_deims_base_url())
```

**Arguments**

deims\_url        A character. DEIMS-SDR base URL. Defaults to package settings.

**Value**

version number.

---

get\_deims\_base\_url *Get DEIMS-SDR base URL*

---

**Description**

Get DEIMS-SDR base URL

**Usage**

```
get_deims_base_url()
```

**Value**

DEIMS-SDR base URL

**See Also**

Other package\_customizable\_settings: [package\\_settings](#)



---

get\_ilter\_envcharacts *Obtain a list of all Environmental Characteristics of ILTER sites.*

---

## Description

**[Questioning]** This function obtains all Environmental Characteristics: title, URI, geo-coordinates, country name, and elevation of all **ILTER sites (more than 1200 around the world)**, through the DEIMS-SDR API.

This function gathers in a unique tibble all the Environmental Characteristics from all ILTER sites. Note that the execution time for this function is very high.

If the objective is obtain information about Environmental Characteristics on a few sites, it is better to use other more specific functions (e.g. `get_network_envcharacts()` or `get_site_info()`) or using other methods (**How to about sites informations**).

## Usage

```
get_ilter_envcharacts(sitesNum = 0)
```

## Arguments

sitesNum      A integer. The number of the sites that are read to get the information. Use this parameter only to get a sample of the output of this function. If the value of sitesNum is 0 (default) all the ILTER sites will be parsed and the waiting time will be long.

## Value

The output of the function is a tibble containing all the Environmental Characteristics of ILTER's sites.

## Author(s)

Alessandro Oggioni, PhD (2020) [oggioni.a@irea.cnr.it](mailto:oggioni.a@irea.cnr.it)

## References

Wickham H, François R, Henry L, Müller K (2022). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.9, <https://CRAN.R-project.org/package=dplyr>.

## Examples

```
## Not run:  
listEnvCharacts <- get_ilter_envcharacts(sitesNum = 10)  
listEnvCharacts[1:20, ]  
  
## End(Not run)
```

---

get\_ilter\_generalinfo *Download information of all ILTER sites or a subset of ILTER sites.*

---

## Description

**[Questioning]** This function downloads generic information of sites of **ILTER sites (more than 1200 around the world)**, through the DEIMS-SDR API. Return a tibble object.

## Usage

```
get_ilter_generalinfo(country_name = NA, site_name = NA, show_map = FALSE)
```

## Arguments

country_name	A character. Country name (complete name in English, French, Italian, German, OR 2 character ISO code) of DEIMS sites to retrieve. Partial matching of country names is NOT supported.
site_name	A character. This character string filters by site name where partial matching is supported. At least one of country_name or site_name must be specified
show_map	A boolean. If TRUE a Leaflet map of site locations is shown. Default FALSE

## Value

An sf object of the bounding boxes of sites in the filtered list, containing the name, DEIMS ID, longitude, latitude, average altitude, and affiliation of the filtered ILTER sites. If no bounding box is available, the centroid is returned.

## Note

at least one of country\_name or site\_name must be specified

## Author(s)

Alessandro Oggioni, PhD (2021) <oggioni.a@irea.cnr.it>  
Micha Silver, PhD (2021) <silverm@post.bgu.ac.il>  
Paolo Tagliolato, PhD (2023) <tagliolato.p@irea.cnr.it>

## References

Ooms J (2014). “The jsonlite Package: A Practical and Consistent Mapping Between JSON Data and R Objects.” *arXiv:1403.2805 [stat.CO]*. <https://arxiv.org/abs/1403.2805>.

Pebesma E (2018). “Simple Features for R: Standardized Support for Spatial Vector Data.” *The R Journal*, **10**(1), 439–446. doi:10.32614/RJ2018009.

Cheng J, Karambelkar B, Xie Y (2022). *leaflet: Create Interactive Web Maps with the JavaScript ‘Leaflet’ Library*. R package version 2.1.1, <https://CRAN.R-project.org/package=leaflet>.

**Examples**

```
## Not run:
# list of the all sites info with ILTER
listOfAllSites <- get_ilter_generalinfo()
length(listOfAllSites[,1])

# example about country name parameter
sitesAustria <- get_ilter_generalinfo(country_name = "Austri")
# (matches Austria, but not Australia)
length(sitesAustria$title)

# example of single site in a country
eisenwurz <- get_ilter_generalinfo(
  country_name = "Austri",
  site_name = "Eisen"
)
eisenwurz[,1:2]
# extract DEIMS.Id
eisenwurz_deimsid <- eisenwurz$uri
eisenwurz_deimsid

# example of single site in a country and return only map
get_ilter_generalinfo(
  country_name = "Italy",
  site_name = "Maggiore",
  show_map = TRUE
)

## End(Not run)
```

---

```
get_ilter_observedProperties
```

*Obtain information about the observed properties collected of all sites  
ILTER.*

---

**Description**

**[Questioning]** Return a tibble object containing observed properties collected by all of the **ILTER sites (more than 1200 around the world)**, available from **DEIMS-SDR**.

This function gathers in a unique tibble all the observed properties from all ILTER sites. Note that the execution time for this function is very high.

If the objective is obtain information about observed properties on a few sites, it is better to use other more specific functions (e.g. `get_network_observedProperties()` or `get_site_info()`) or using other methods (**How to about sites informations**).

**Usage**

```
get_ilter_observedProperties(sitesNum = 0)
```

**Arguments**

sitesNum            A integer. The number of the sites that are read to get the information. Use this parameter only to sample the output of this function. If the value of sitesNum is #' 0 (default) all the ILTER sites will be parsed and the waiting time will be long.

**Value**

The output of the function is a tibble containing the list of observed properties and their URI (Uniform Resource Identifier) collected in all ILTER sites.

**Author(s)**

Alessandro Oggioni, PhD (2020) <oggioni.a@irea.cnr.it>

**References**

Ooms J (2014). "The jsonlite Package: A Practical and Consistent Mapping Between JSON Data and R Objects." *arXiv:1403.2805 [stat.CO]*. <https://arxiv.org/abs/1403.2805>.

Wickham H, François R, Henry L, Müller K (2022). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.9, <https://CRAN.R-project.org/package=dplyr>.

**Examples**

```
## Not run:
listParams <- get_ilter_observedProperties(sitesNum = 20)
listParams[1:10, ] %>%
  dplyr::rows_insert(
  dplyr::tibble(
    parameterLabel = "...", parameterUri = "..."
  )
)
## End(Not run)
```

---

```
get_ilter_research_topics
```

*Obtain the information about the Research Topics of ILTER sites.*

---

**Description**

**[Defunct]** This function was defunct because the section about research topics of the site in DEIMS-SDR API version 1.1 has been removed.

This function obtains Research Topics as collected by all **ILTER sites** (more than 1200 around the world), as stored in **DEIMS-SDR**. Note that the execution time for this function is very high.

If the objective is to obtain information about Research Topics from a few sites, it is better to use other more specific functions (e.g. `get_network_research_topics()` or `get_site_info()`) or using other methods (**How to about sites informations**).

**Usage**

```
get_ilter_research_topics(sitesNum = NULL)
```

**Arguments**

`sitesNum` A integer. It is the number of the sites that are read to get the information. Use this parameter only to get an example of the output of this function. If the value of `sitesNum` is 0 (default) all the ILTER sites will be parsed and the waiting time will be long.

**Value**

The output of the function is a tibble containing the research topics and their URI (Uniform Resource Identifier) of all ILTER sites.

**Author(s)**

Alessandro Oggioni, PhD (2020) <oggioni.a@irea.cnr.it>

**References**

Ooms J (2014). "The jsonlite Package: A Practical and Consistent Mapping Between JSON Data and R Objects." *arXiv:1403.2805 [stat.CO]*. <https://arxiv.org/abs/1403.2805>.

Wickham H, François R, Henry L, Müller K (2022). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.9, <https://CRAN.R-project.org/package=dplyr>.

**Examples**

```
## Not run:
listResearchTopics <- get_ilter_research_topics(sitesNum = 20)
listResearchTopics[1:10, ] %>%
  dplyr::rows_insert(
    dplyr::tibble(
      researchTopicsLabel = "...",
      researchTopicsUri = "...")
  )
## End(Not run)
```

---

get\_network\_envcharacts

*Obtain a list of all Environmental Characteristics of sites in an eLTER Network.*

---

## Description

**[Stable]** This function obtains all Environmental Characteristics: title, URI, geo-coordinates, country name, and elevation of eLTER Network sites (e.g. [LTER- Italy network](#)), through the DEIMS-SDR API.

## Usage

```
get_network_envcharacts(networkDEIMSID)
```

## Arguments

`networkDEIMSID` A character. DEIMS ID of network from DEIMS-SDR website. DEIMS ID information [here](#) and Complete list of ILTER networks [here](#). The DEIMS ID of network is the URL for the network page.

## Value

The output of the function is a tibble containing all the Environmental Characteristics of the network's sites.

## Author(s)

Alessandro Oggioni, PhD (2020) <oggioni.a@irea.cnr.it>

## References

Ooms J (2014). "The jsonlite Package: A Practical and Consistent Mapping Between JSON Data and R Objects." *arXiv:1403.2805 [stat.CO]*. <https://arxiv.org/abs/1403.2805>.

Wickham H, François R, Henry L, Müller K (2022). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.9, <https://CRAN.R-project.org/package=dplyr>.

## Examples

```
## Not run:
listEnvCharacts <- get_network_envcharacts(
  networkDEIMSID =
    "https://deims.org/network/7fef6b73-e5cb-4cd2-b438-ed32eb1504b3"
)
listEnvCharacts[1:10, ]

## End(Not run)
```

---

`get_network_observedProperties`

*Obtain a list of all the observed properties of sites in an eLTER Network.*

---

## Description

**[Stable]** This function obtains all observed properties collected in an eLTER Network (e.g. [LTER-Italy network](#)), through the DEIMS-SDR API.

## Usage

```
get_network_observedProperties(networkDEIMSID)
```

## Arguments

`networkDEIMSID` A character. The DEIMS ID of network from DEIMS-SDR website. DEIMS ID information [here](#) and Complete list of ILTER networks [here](#). The DEIMS ID of network is the URL for the network page.

## Value

The output of the function is a tibble containing the list of observed properties and their URI (Uniform Resource Identifier) collected by the network's sites.

## Author(s)

Alessandro Oggioni, PhD (2020) <oggioni.a@irea.cnr.it>

## References

Ooms J (2014). "The jsonlite Package: A Practical and Consistent Mapping Between JSON Data and R Objects." *arXiv:1403.2805 [stat.CO]*. <https://arxiv.org/abs/1403.2805>.

Wickham H, François R, Henry L, Müller K (2022). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.9, <https://CRAN.R-project.org/package=dplyr>.

## Examples

```
## Not run:
listParams <- get_network_observedProperties(
  networkDEIMSID =
    "https://deims.org/networks/e0f680c2-22b1-4424-bf54-58aa9b7476a0"
)
listParams[1:10, ] %>%
dplyr::rows_insert(
  dplyr::tibble(parameterLabel = "...", parameterUri = "...")
)
```

```
## End(Not run)
```

---

```
get_network_related_resources
```

*Obtain information about the Related Resources (dataset and activity) of a eLTER Network.*

---

## Description

**[Stable]** This function obtains the Related Resources information (title and URL), as a stored in [DEIMS-SDR catalogue](#), of all eLTER sites belonging to an eLTER Network (e.g. [LTER Italy network](#)).

## Usage

```
get_network_related_resources(networkDEIMSID)
```

## Arguments

`networkDEIMSID` A character. It is the DEIMS ID of network make from DEIMS-SDR website. DEIMS ID information [here](#) and Complete list of ILTER networks [here](#). The DEIMS ID of network is the URL for the network page.

## Value

The output of the function is a tibble containing the related resources shared by the network's sites.

## Author(s)

Alessandro Oggioni, PhD (2020) <oggioni.a@irea.cnr.it>

## References

Ooms J (2014). "The jsonlite Package: A Practical and Consistent Mapping Between JSON Data and R Objects." *arXiv:1403.2805 [stat.CO]*. <https://arxiv.org/abs/1403.2805>.

Wickham H, François R, Henry L, Müller K (2022). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.9, <https://CRAN.R-project.org/package=dplyr>.

## Examples

```
## Not run:
listRelatedResources <- get_network_related_resources(
  networkDEIMSID =
    "https://deims.org/network/7fef6b73-e5cb-4cd2-b438-ed32eb1504b3"
)
listRelatedResources
```



```
## End(Not run)
```

---

```
get_network_research_topics
```

*Obtain a list of Research Topics handled in an eLTER Network.*

---

### Description

**[Defunct]** This function was defunct because the section about research topics of the site in DEIMS-SDR API version 1.1 has been removed.

This function obtains Research Topics collected by all of the eLTER sites belonging to an eLTER Network (e.g. [LTER- Italy network](#)), as a stored into [DEIMS-SDR](#).

### Usage

```
get_network_research_topics(networkDEIMSID)
```

### Arguments

`networkDEIMSID` A character. The DEIMS ID of a network from DEIMS-SDR website. DEIMS ID information [here](#) and Complete list of ILTER networks [here](#). The DEIMS ID of network is the URL for the network page.

### Value

The output of the function is a tibble containing the research topics and their URI (Uniform Resource Identifier) collected by network's sites.

### Author(s)

Alessandro Oggioni, PhD (2020) <oggioni.a@irea.cnr.it>

### References

Ooms J (2014). "The jsonlite Package: A Practical and Consistent Mapping Between JSON Data and R Objects." *arXiv:1403.2805 [stat.CO]*. <https://arxiv.org/abs/1403.2805>.

Wickham H, François R, Henry L, Müller K (2022). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.9, <https://CRAN.R-project.org/package=dplyr>.

## Examples

```
## Not run:
listResearchTopics <- get_network_research_topics(
  networkDEIMSID =
    "https://deims.org/network/7fef6b73-e5cb-4cd2-b438-ed32eb1504b3"
)
listResearchTopics[1:10, ] %>%
  dplyr::rows_insert(
    dplyr::tibble(
      researchTopicsLabel = "...",
      researchTopicsUri = "..."
    )
  )
## End(Not run)
```

---

get\_network\_sites      *Retrieve a list of sites in an eLTER Network.*

---

## Description

**[Stable]** This function return a spatial point vector object including title, date late updated, URI, and coordinates, stored in [DEIMS-SDR catalogue](#), of all the eLTER sites belonging to an eLTER Network (e.g. [LTER- Italy network](#)).

## Usage

```
get_network_sites(networkDEIMSID)
```

## Arguments

networkDEIMSID A character. The DEIMS ID of the network from DEIMS-SDR website. DEIMS ID information [here](#) and Complete list of networks [here](#). The DEIMS ID of network is the URL for the network page.

## Value

The output of the function is a point vector of sf class (package sf) of the network's sites.

## Author(s)

Alessandro Oggioni, PhD (2020) <oggioni.a@irea.cnr.it>

## References

Wickham H (2022). *httr: Tools for Working with URLs and HTTP*. R package version 1.4.3, <https://CRAN.R-project.org/package=httr>.

Wickham H, François R, Henry L, Müller K (2022). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.9, <https://CRAN.R-project.org/package=dplyr>.

Ooms J (2014). “The jsonlite Package: A Practical and Consistent Mapping Between JSON Data and R Objects.” *arXiv:1403.2805 [stat.CO]*. <https://arxiv.org/abs/1403.2805>.

Pebesma E (2018). “Simple Features for R: Standardized Support for Spatial Vector Data.” *The R Journal*, **10**(1), 439–446. doi:10.32614/RJ2018009.

Cheng J, Karambelkar B, Xie Y (2022). *leaflet: Create Interactive Web Maps with the JavaScript ‘Leaflet’ Library*. R package version 2.1.1, <https://CRAN.R-project.org/package=leaflet>.

## Examples

```
## Not run:
# The sites of LTER-Italy network
listSites <- get_network_sites(
  networkDEIMSID =
    "https://deims.org/network/7fef6b73-e5cb-4cd2-b438-ed32eb1504b3"
)
listSites

# The sites of LTER Europe network
euSites <- get_network_sites(
  networkDEIMSID =
    "https://deims.org/networks/4742ffca-65ac-4aae-815f-83738500a1fc"
)
euSites

## End(Not run)
```

---

get\_sensor\_info

*Obtain the information about of an eLTER sensor.*

---

## Description

**[Experimental]** This function obtains the information about of an eLTER sensor (e.g. <https://deims.org/sensors/3845475c-4aec-4dd7-83b4-0ab6ba95db35>) provided in **DEIMS-SDR catalogue**.

## Usage

```
get_sensor_info(sensorid, show_map = FALSE)
```

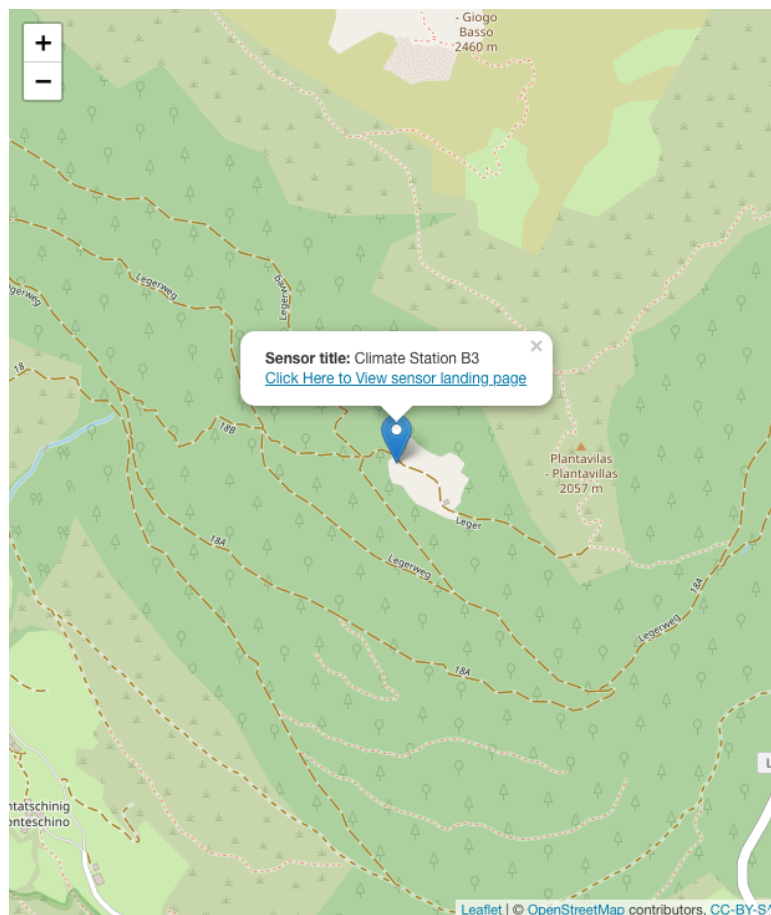
### Arguments

sensorid	A character. It is the DEIMS ID of sensor make from DEIMS-SDR website. DEIMS ID information <a href="#">here</a> . The DEIMS.iD of sensor is the URL for the sensor page.
show_map	A boolean. If TRUE a Leaflet map with occurrences is shown. Default FALSE.

### Value

The output of the function is a tibble with main features of the activities in a site, and a leaflet map plot.

### The function output



### Author(s)

Alessandro Oggioni, PhD (2020) <oggioni.a@irea.cnr.it>

## Examples

```
# only table of sensor information
sensor_B3 <- get_sensor_info(
  sensorid =
    "https://deims.org/sensors/3845475c-4aec-4dd7-83b4-0ab6ba95db35",
  show_map = TRUE
)
sensor_B3

# print the map of the sensor
Licor <- get_sensor_info(
  sensorid =
    "https://deims.org/sensors/4a7ad644-f2e7-4224-965b-ec5ef5365655",
  show_map = FALSE
)
Licor

# Moldaenke FluoroProbe sensor
sensor_FP <- get_sensor_info(
  sensorid = "https://deims.org/sensors/82635223-a4f4-498c-b283-9c95999d9d2f",
  show_map = FALSE
)
sensor_FP
```

---

get\_sensor\_observed\_properties

*Obtain the observed properties measured by a sensor.*

---

## Description

**[Experimental]** This function obtains the observed properties by procedure/sensor through Sensor Observation Service (SOS).

## Usage

```
get_sensor_observed_properties(sosURL, procedure)
```

## Arguments

sosURL	A character. The endpoint of the Sensor Observation Service (SOS) service.
procedure	A character. It is a procedure/sensor ID.

## Value

The output of the function is a tibble with the labels and URI (Uniform Resource Identifier) of each observed property, the code and URI of Units Of Measurement (UOM) of the observed properties as declared in the Sensor Observation Service (SOS). Codes and URIs as stated in [QUDT.org](http://qudt.org)

are also present. QUDT is a public charity nonprofit organization founded to provide semantic specifications for units of measure, quantity kind, dimensions and data types. NB this function returns a valued string only in the case where the UOM refers to a NERC vocabulary term (e.g. <http://vocab.nerc.ac.uk/collection/P06/current/UPAA/> for °C).

### Author(s)

Alessandro Oggioni, PhD <oggioni.a@irea.cnr.it>

### Examples

```
## Not run:
FP <- get_sensor_observed_properties(
  sosURL = "http://getit.lteritalia.it/observations/service",
  procedure = "http://www.get-it.it/sensors/getit.lteritalia.it/procedure/noOwnerDeclared/noModelDeclared/noSeriesDeclared"
)
FP

eurac_monalisa <- get_sensor_observed_properties(
  sosURL = "http://monalisasos.eurac.edu/sos/service",
  procedure = "QuantumSensor_nemef2000"
)
eurac_monalisa

obsProsAir <- get_sensor_observed_properties(
  sosURL = "http://getit.lteritalia.it/observations/service",
  procedure = "http://www.get-it.it/sensors/getit.lteritalia.it/procedure/noOwnerDeclared/noModelDeclared/noSeriesDeclared"
)
obsProsAir

NIVA <- get_sensor_observed_properties(
  sosURL = "https://hydro-sos.niwa.co.nz/",
  procedure = "Hydrometric_Station"
)
NIVA

## End(Not run)
```

---

get\_site\_info

*Obtain details about an eLTER site.*

---

### Description

**[Stable]** This function obtains information of a single eLTER site, as a stored in **DEIMS-SDR catalogue**, through the DEIMS-SDR API.

### Usage

```
get_site_info(deimsid, category = NA)
```

## Arguments

deimsid	A character. The DEIMS ID of the site from DEIMS-SDR website. DEIMS ID information <a href="#">here</a> .
category	A category. This parameter selects which category or categories are retrieved and returned in the result. Possible value are: 'Affiliations', 'Boundaries', 'Contacts', 'EnvCharacts', 'General', 'Infrastructure', 'observedProperties', 'RelateRes'. Multiple values can be indicated.

## Value

The output of the function is a tibble with main features of the site and the selected information, such as: networks and projects in which the site is involved. If category 'Boundaries' is indicated an sf object is returned

## Author(s)

Alessandro Oggioni, PhD (2020) <oggioni.a@irea.cnr.it>

## References

Wickham H, François R, Henry L, Müller K (2022). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.9, <https://CRAN.R-project.org/package=dplyr>.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.

Pebesma E (2018). "Simple Features for R: Standardized Support for Spatial Vector Data." *The R Journal*, **10**(1), 439–446. doi:10.32614/RJ2018009.

Cheng J, Karambelkar B, Xie Y (2022). *leaflet: Create Interactive Web Maps with the JavaScript 'Leaflet' Library*. R package version 2.1.1, <https://CRAN.R-project.org/package=leaflet>.

## Examples

```
site <- get_site_info(  
  deimsid = "https://deims.org/f30007c4-8a6e-4f11-ab87-569db54638fe",  
  category = "Boundaries"  
)  
site  
  
siteInfo <- get_site_info(  
  deimsid = "https://deims.org/f30007c4-8a6e-4f11-ab87-569db54638fe",  
  category = c("EnvCharacts", "Affiliations")  
)  
siteInfo
```

---

get\_site\_MODIS                      *Acquire a time series of MODIS satellite products*

---

### Description

**[Stable]** Acquire either Land Surface Temperature (LST) or Vegetation Index (NDVI) both cropped to an eLTER site boundary. Download a timeseries of MODIS images containing the requested product and optionally:

Plot a time series graph of the average values over the site.

Create and show an aggregated map of the acquired product

Use of this function requires registering on the EarthData website:

<https://urs.earthdata.nasa.gov/home> In order to guard your user credentials, please save your user-name and password to environment variables. i.e.

```
Sys.setenv("earthdata_user"="homer_simpson") Sys.setenv("earthdata_pass"="bart&lucy")
```

### Usage

```
get_site_MODIS(
  deimsid,
  product = "VI",
  from_date = "2010.01.01",
  to_date = "2020.31.12",
  output_dir = NULL,
  plot_ts = TRUE,
  output_proj = "3035",
  download_range = "Full",
  show_map = FALSE
)
```

### Arguments

deimsid	character. The DEIMS ID of the site from DEIMS-SDR website. DEIMS ID information <a href="#">here</a> .
product	character. The requested product. One of: "LST", "VI", "ET", "LAI". "LST" for Land Surface Temperature, night and day, 8 day intervals at 1000m resolution "VI" for Vegetation Indices, NDVI and EVI 16 day intervals at 250m resolution "LAI" for Leaf area index and FPAR at 500m resolution "ET" for Evapotranspiration, 8 day interval at 500m resolution Default is "VI".
from_date	character: the start date formatted as YYYY.MM.DD
to_date	character: the end date formatted as YYYY.MM.DD
output_dir	character: where to save downloaded rasters (Default is tempdir())
plot_ts	boolean: whether to plot the time series, Default TRUE.
output_proj	character: the EPSG code of desired output projection. Default is "3035", the European LAEA coordinate reference system.



download_range	character: one of "Full" or "Seasonal". Specifies whether to acquire all images between start and end dates, or only for a specific season. e.g. if the starting date is "2010.01.01" and the ending date is "2020.02.28" then only images for January and February are acquired, over the 10 year time span. (See example)
show_map	character: Whether to create, save and display an aggregated map from the time series of acquired MODIS products. See note below. This string must be one of:  FALSE (the default): no map is shown or created. Otherwise: an aggregation function such as "mean", "max", or "min."

### Details

Certain layers from each of the supported MODIS products are acquired.

- from: "LST\_3band\_emissivity\_8day\_1km (M\*D21A2)" two "Land surface temperature" bands are acquired:  
"LST\_Day\_1KM", "LST\_Night\_1KM"
- from: "Vegetation Indexes\_16Days\_250m (M\*D13Q1)" two Vegetation Indices are acquired:  
"NDVI" and "EVI"
- from: "LAI\_8Days\_500m (M\*D15A2H)" two indices are acquired:  
"Fpar" and "Lai"
- from: "Net\_ET\_8Day\_500m (M\*D16A2)" one Evapotranspiration band:  
"PET" (Potential EvapoTranspiration)

### NOTES:

- The default output\_dir is tempdir(), so the downloaded MODIS files will be deleted when exiting R. Enter a permanent path for output\_dir to save the files.
- Use the plot\_ts parameter to create and save line plots of a time series of average pixel values over the site.
- Use the show\_map parameter to create and show a time series aggregation map of the product over the site.
- Evapotranspiration products are available only up to 2018
- Plotting with show\_map requires: packageVersion("leaflet")>"2.1.1"

### Value

Full path of all downloaded and cropped Geotiff files

### Author(s)

Micha Silver, PhD (2020) <silverm@post.bgu.ac.il>

Alessandro Oggioni, PhD (2020) <oggioni.a@irea.cnr.it>

## References

Busetto L, Ranghetti L (2016). “MODISstp: an R package for preprocessing of MODIS Land Products time series.” *Computers & Geosciences*, **97**, 40-48. ISSN 0098-3004, doi:10.1016/j.cageo.2016.08.020, <https://github.com/ropensci/MODISstp>.

Pebesma E (2018). “Simple Features for R: Standardized Support for Spatial Vector Data.” *The R Journal*, **10**(1), 439–446. doi:10.32614/RJ2018009.

Hijmans RJ (2022). *terra: Spatial Data Analysis*. R package version 1.5-21, <https://CRAN.R-project.org/package=terra>.

MODIS images from: <https://lpdaac.usgs.gov>, maintained by the NASA EOSDIS Land Processes Distributed Active Archive Center (LP DAAC) at the USGS Earth Resources Observation and Science (EROS) Center, Sioux Falls, South Dakota. 2018, <https://lpdaac.usgs.gov/resources/data-action/aster-ultimate-2018-winter-olympics-observer/>.

## Examples

```
## Not run:
# Lago Maggiore - Italy, LST over an 6 month time span
# Saved in LAEA ETRS89 coordinate reference system
# This example completes in about 10 mins
deimsid = "https://deims.org/f30007c4-8a6e-4f11-ab87-569db54638fe"
product <- "LST"
from_date <- "2018.03.01"
to_date <- "2018.08.30"
output_dir <- tempdir()
output_proj <- "3035"
download_list <- ReLTER::get_site_MODIS(deimsid,
  product = product,
  from_date = from_date, to_date = to_date,
  output_dir = output_dir,
  plot_ts = TRUE,
  output_proj = output_proj)

# Northern Negev LTER - Israel, NDVI over 4 winter months,
# projected to Israeli 05/12 CRS
# This example completes in about 30 mins
deimsid <- "https://deims.org/871a90b2-e372-456a-93e3-518ad1e11239"
from_date <- "2018.01.01"
to_date <- "2018.04.30"
product = "VI"
output_dir <- tempdir()
output_proj <- "6991"
download_list <- ReLTER::get_site_MODIS(deimsid,
  product = product,
  from_date = from_date, to_date=to_date,
  output_dir = output_dir,
  plot_ts = TRUE,
  output_proj = output_proj)

# Nationalpark Mols Bjerger - Denmark, 10 year only for July
# Show aggregated mean NDVI and EVI, (No time series plot)
```

```
# projected to EPSG:25832 (UTM zone 32, ETRS89)
# Takes about 3/4 hour to run...
deimsid <- "https://deims.org/8407da23-d75d-4a02-a5a5-7b9701a86743"
from_date <- "2005.07.01"
to_date <- "2015.08.01"
output_dir <- tempdir()
output_proj <- "25832"
product <- "VI"
download_list <- ReLTER::get_site_MODIS(deimsid,
  product = product,
  from_date = from_date, to_date = to_date,
  output_dir = output_dir,
  output_proj = output_proj,
  download_range = "Seasonal",
  plot_ts = FALSE,
  show_map = "mean")

# Braila Islands - Romania, 2 year time series of evapotranspiration
# projected to Pulkova 1942(59) Zone 9 CRS, EPSG:3839
# Takes almost 1.5 hours to run (requires 2 MODIS tiles)
deimsid <- "https://deims.org/d4854af8-9d9f-42a2-af96-f1ed9cb25712"
from_date <- "2015.01.01"
to_date <- "2016.12.31"
output_dir <- tempdir()
output_proj <- "3839"
product <- "ET"
download_list <- ReLTER::get_site_MODIS(deimsid,
  product = product,
  from_date = from_date, to_date = to_date,
  output_dir = output_dir,
  output_proj = output_proj,
  download_range = "Full",
  plot_ts = TRUE,
  show_map = FALSE)

# Gran Paradiso National Park - Italy,
# 1 year time series of LAI and aggregated map
# projected to ETRS89 LAEA, EPSG:3035
# Takes about 3/4 hour to run
deimsid <- "https://deims.org/e33c983a-19ad-4f40-a6fd-1210ee0b3a4b"
from_date <- "2020.01.01"
to_date <- "2020.12.31"
output_dir <- tempdir()
output_proj <- "3035"
product <- "LAI"
download_list <- ReLTER::get_site_MODIS(deimsid,
  product = product,
  from_date = from_date, to_date = to_date,
  output_dir = output_dir,
  output_proj = output_proj,
  download_range = "Full",
```

```

plot_ts = TRUE,
show_map = "mean")

## End(Not run)

```

---

get_site_ODS	<i>Acquire various raster layers from <a href="https://maps.opendatascience.eu/ODS">https://maps.opendatascience.eu/ODS</a> Europe and crops to an eLTER site boundary.</i>
--------------	---

---

## Description

**[Stable]** Download and return a SpatRaster object containing the requested dataset from **ODS**, cropped to an eLTER site boundary, which is obtained from the DEIMS-SDR API.

## Usage

```
get_site_ODS(deimsid, dataset = "landcover")
```

## Arguments

deimsid	A character. The DEIMS ID of the site from DEIMS-SDR website. DEIMS ID information <a href="#">here</a> .
dataset	A character. The requested dataset. One of: "landcover", "clc2018", "osm_buildings", "natura2000", "ndvi_spring", "ndvi_summer", "ndvi_autumn", "ndvi_winter", "ndvi_trend", "forest_broadleaf", "forest_mixed", "forest_coniferous". Default is "landcover".

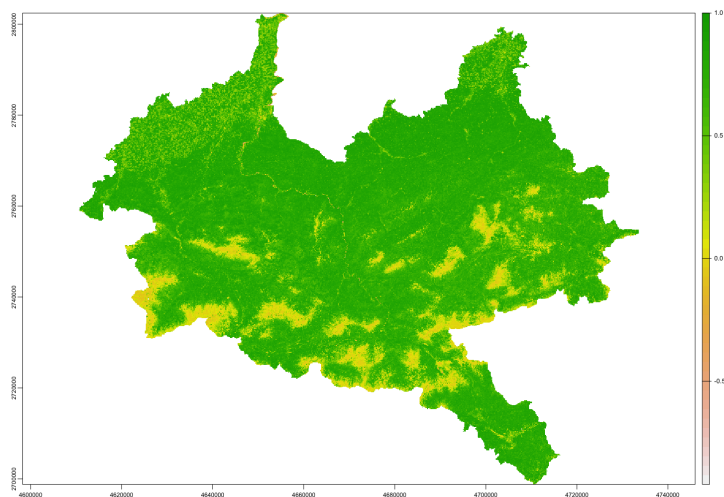
## Details

Supported datasets from the ODS repository include: Landcover: Land-cover class according to the highest probability, generated by a spatiotemporal ensemble-ML model. 30 m. resolution CLC2018: Corine land cover rasterized to 100m spatial resolution and provided by Copernicus Land Monitoring Service. OSM buildings: Buildings according to OSM polygons and the Copernicus impervious build-up layer (2018), aggregated and rasterized first to 10m spatial resolution and after downsampled to 30m by spatial average. Natura2000: Protected areas rasterized from NATURA 2000 (A, B and C site categories) and OSM (IUCN Ia, IUCN Ib, IUCN 2, IUCN 3, IUCN 4, IUCN 5, IUCN 6 and others categories), first to 10m spatial resolution and after downsampled to 30m by spatial average. The overlap areas are indicated in a new category. NDVI: NDVI time-series, derived from the Landsat quarterly temporal composites NDVI Trend from 2000 - 2019 as OLS regression Forests: Broadleaf, coniferous or mixed forests All datasets are georeferenced to the EPSG:3035 coordinate reference system. and all except clc2018 have 30 meters resolution

## Value

The function returns a `SpatRaster` object (from the `terra` package) of the requested dataset, cropped to the site boundaries. The user should save the raster to disk, if necessary. i.e. `writeRaster(ds_site, "site_dataset.tif")`

## The function output



## Author(s)

Micha Silver, PhD (2020) <silverm@post.bgu.ac.il>

Alessandro Oggioni, PhD (2020) <oggioni.a@irea.cnr.it>

## References

Wickham H, François R, Henry L, Müller K (2022). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.9, <https://CRAN.R-project.org/package=dplyr>.

Pebesma E (2018). "Simple Features for R: Standardized Support for Spatial Vector Data." *The R Journal*, **10**(1), 439–446. doi:10.32614/RJ2018009.

Hijmans RJ (2022). *terra: Spatial Data Analysis*. R package version 1.5-21, <https://CRAN.R-project.org/package=terra>.

## Examples

```
## Not run:  
# Landcover for Angelo Mosso  
siteLandcover <- get_site_ODS(  
  deimsid = "https://deims.org/17210eba-d832-4759-89fa-9ff127cbdf6e",  
  dataset = "landcover"  
)  
siteLandcover  
terra::plot(siteLandcover)
```

```
# NDVI for Eisenwurz
siteNDVI <- get_site_ODS(
  deimsid = "https://deims.org/d0a8da18-0881-4ebe-bccf-bc4cb4e25701",
  dataset = "ndvi_summer"
)
siteNDVI
terra::plot(siteNDVI)

## End(Not run)
```

---

```
get_site_speciesOccurrences
```

*Trims by eLTER site the species occurrence from different sources*

---

### Description

**[Stable]** This function acquires data from GBIF <https://www.gbif.org> (via rgbif), iNaturalist <https://www.inaturalist.org/> and OBIS <https://obis.org/> and crops to an eLTER site boundary, which is obtained from the DEIMS-SDR sites API.

### Usage

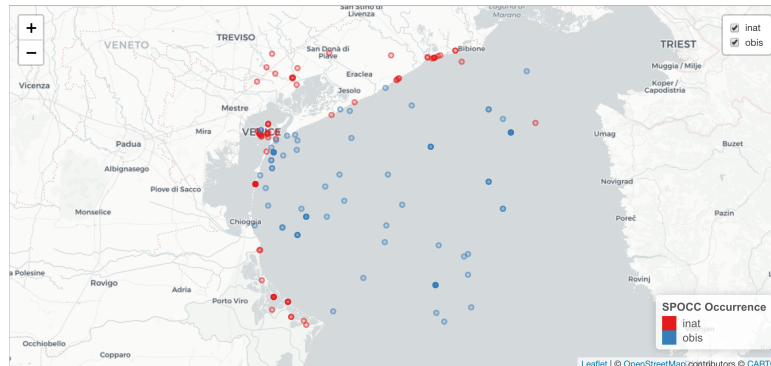
```
get_site_speciesOccurrences(
  deimsid,
  list_DS,
  show_map = FALSE,
  limit = 500,
  exclude_inat_from_gbif = TRUE
)
```

### Arguments

deimsid	A character. The DEIMS.iD of the site from DEIMS-SDR website. DEIMS.iD information <a href="#">here</a> .
list_DS	A character. Data source to get data from, any combination of gbif, inat and/or obis.
show_map	A boolean. If TRUE a Leaflet map with occurrences is shown. Default FALSE.
limit	A numeric. Number of records to return. This is passed across all sources. Default: 500 for each source. BEWARE: if you have a lot of species to query for (e.g., n = 10), that's 10 * 500 = 5000, which can take a while to collect. So, when you first query, set the limit to something smallish so that you can get a result quickly, then do more as needed.
exclude_inat_from_gbif	A boolean. If TRUE, when list_DS contains both "gbif" and "inat", filter out gbif records originating from iNaturalist (in order to avoid duplicates). Default TRUE.

**Value**

The output of the function is a list of sf one for each of the repositories specified in the list\_DS parameter.

**The function output****Author(s)**

Alessandro Oggioni, PhD (2020) <oggioni.a@irea.cnr.it>

Paolo Tagliolato, PhD (2020) <tagliolato.p@irea.cnr.it>

Martina Zilioli <zilioli.m@irea.cnr.it>

**Examples**

```
## Not run:
# terrestrial site Saldur River Catchment
occ_SRC <- get_site_speciesOccurrences(
  deimsid =
    "https://deims.org/97ff6180-e5d1-45f2-a559-8a7872eb26b1",
  list_DS = c("gbif", "inat"),
  show_map = FALSE,
  limit = 10
)
occ_SRC

# marine site Gulf of Venice only obis
occ_GoV <- get_site_speciesOccurrences(
  deimsid =
    "https://deims.org/758087d7-231f-4f07-bd7e-6922e0c283fd",
  list_DS = "obis",
  show_map = FALSE,
  limit = 10
)
occ_GoV

# marine site Gulf of Venice, all repositories are invoked
# gbif, inat and obis
```

```

occ_GoV_all <- get_site_speciesOccurrences(
  deimsid =
    "https://deims.org/758087d7-231f-4f07-bd7e-6922e0c283fd",
  list_DS = c("gbif", "inat", "obis"),
  show_map = TRUE,
  limit = 10
)
occ_GoV_all

## End(Not run)

```

---

get\_sos\_foi

*Obtain the information about the feature of interest.*


---

## Description

**[Experimental]** This function obtains the information about Feature(s) Of Interest (FOI(s)) provided by a Sensor Observation Services (SOS).

## Usage

```
get_sos_foi(sosURL, show_map = FALSE)
```

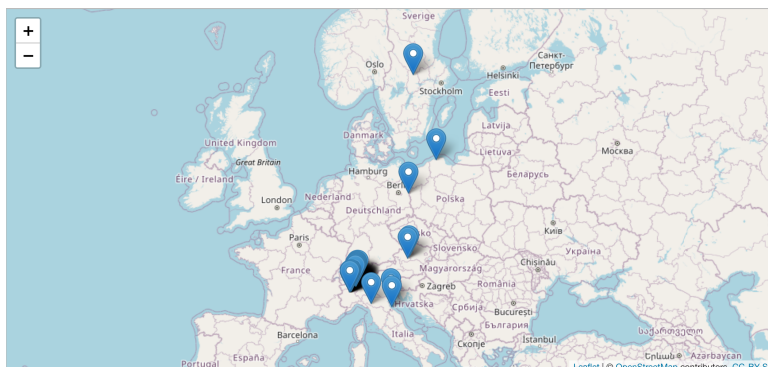
## Arguments

sosURL	A character. The endpoint of the Sensor Observation Service (SOS) service.
show_map	A logical. When TRUE the boundary will be plotted on a Leaflet map. Default FALSE.

## Value

The output of the function is a `sf`. The table contains all the information about Feature(s) Of Interest (FOI(s)) provided by a Sensor Observation Services (SOS). The columns are about: sampling feature (typeSf, description, name), sampled feature id (sampledFeature), coordinate reference system code (srsName) and coordinates (geometry).



**The function output****Author(s)**

Alessandro Oggioni, PhD <oggioni.a@irea.cnr.it>

Paolo Tagliolato, PhD <tagliolato.p@irea.cnr.it>

**Examples**

```
## Not run:
```

```
## Not run:
```

```
LTERItaly <- get_sos_foi(
  sosURL = "http://getit.lteritalia.it/observations/service",
  show_map = TRUE
)
LTERItaly
```

```
eurac_monalisa <- get_sos_foi(
  sosURL = "http://monalisasos.eurac.edu/sos/service",
  show_map = FALSE
)
eurac_monalisa
```

```
NIVA <- get_sos_foi(
  sosURL = "https://hydro-sos.niwa.co.nz/",
  show_map = FALSE
)
NIVA
```

```
## End(Not run)
```

```
## End (Not run)
```

---

get_sos_obs	<i>Obtain the observations from a Sensor Observation Service (SOS).</i>
-------------	---

---

## Description

**[Experimental]** This function obtains the observations shared by Sensor Observation Service (SOS).

## Usage

```
get_sos_obs(sosURL, procedure, foi = NULL, show_map = FALSE)
```

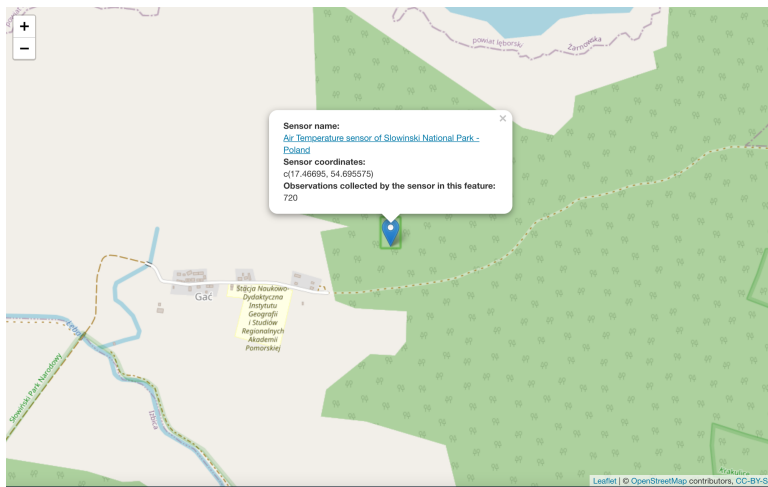
## Arguments

sosURL	A character. The endpoint of the Sensor Observation Service (SOS) service.
procedure	A character. The procedure/sensor ID. Possible value are the Feature of Interest (FOI) ID, which can be obtained via the <code>get_sos_procedure_info()</code> function.
foi	A character. This parameter selects which Feature(s) Of Interest (FOI(s)) are retrieved and returned in the observations. Possible value are the FOI(s) ID, which can be obtained via the <code>get_sos_foi()</code> function. If the FOI parameter is not set, all observations reached in all FOIs by a sensor, will be downloaded. Note that the request time may be very long! Multiple values can be indicated. Default NA.
show_map	A logical. When TRUE the boundary will be plotted on a Leaflet map. Default FALSE.

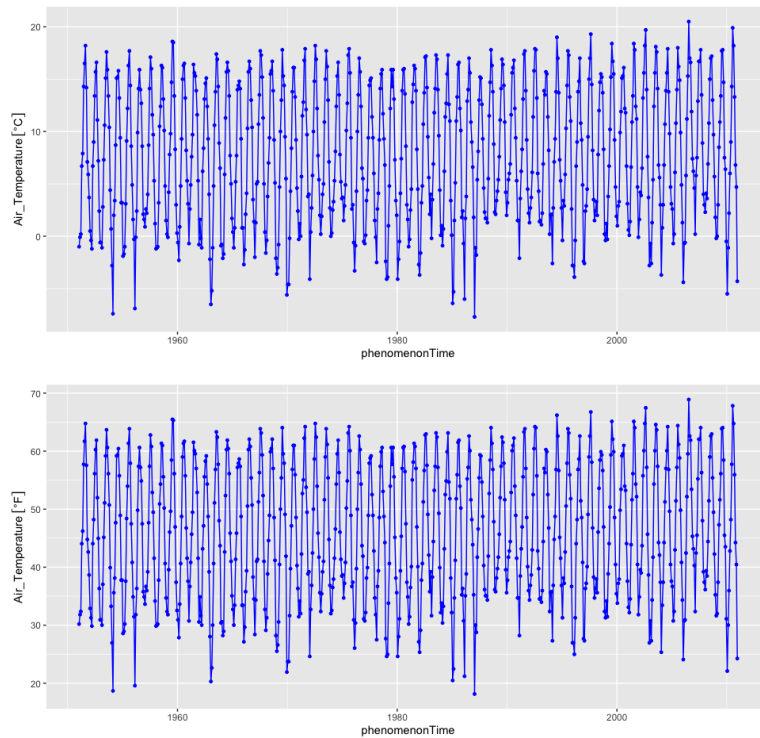
## Value

The output of the function is a tibble. The output can be divided in two parts: the columns concerning the observations and the ancillary information concerning who and where the observations were made. The first part, usually starting with date and time values, contains all columns representing all the observed properties (e.g. air temperature) measured by sensor. The second part contains columns about: Feature(s) Of Interest - FOI (foiLabel and foiID), identifier of the observations block (obsBlockID), procedure/sensor (procedureID and procedureName), sampling feature (typeSf, description, name), sampled feature id (sampledFeature), coordinate reference system code (srsName) and coordinates (lon and lat). A map can be obtained indicating the parameter `show_map` TRUE. The output contains also a semantic link, as provided by SOS. The `uri` attribute contains all URIs of the terms indicated in the headers columns. To the observed properties columns are labeled with a unit of measurement, as mentioned in the SOS, using R package `units`. This labelling simplify the propagation, conversion and derivation of units of collected observed properties.

### The function output



### Possible output from function result



### Author(s)

Alessandro Oggioni, PhD <oggioni.a@irea.cnr.it>

Paolo Tagliolato, PhD <tagliolato.p@irea.cnr.it>

**Examples**

```

## Not run:
## Not run:

# Fluoro Probe sensor
# FP <- get_sos_obs(
#   sosURL = "http://getit.lteritalia.it/observations/service",
#   procedure = "http://www.get-it.it/sensors/getit.lteritalia.it/procedure/noOwnerDeclared/noModelDeclared/noSe
#   foi = c("http://www.get-it.it/sensors/getit.lteritalia.it/sensors/foi/SSF/SP/4326/45.9547/8.63403"),
#   show_map = TRUE
# )
# FP

# Air temperature sensor
airTemp <- get_sos_obs(
  sosURL = "http://getit.lteritalia.it/observations/service",
  procedure = "http://www.get-it.it/sensors/getit.lteritalia.it/procedure/noOwnerDeclared/noModelDeclared/noSer
  show_map = TRUE
)
airTemp

# about units of measurement (UOM)
# the UOM of this observed property is °C
airTemp$Air_Temperature

# is is easily convert to °F
units::set_units(airTemp$Air_Temperature, "°F")

# about semantic enrichment
# the URI of the label of first two columns
# of `airTemp`
attributes(airTemp)$uri

## End(Not run)
## End (Not run)

```

---

get\_sos\_procedurelist *List the procedures of a Sensor Observations Service (SOS).*

---

**Description**

**[Experimental]** Return a list of procedures (e.g. method, algorithm, instrument, sensor, or system which may be used in making observations) store into a **SOS (Sensor Observations Service OGC)**.

**Usage**

```
get_sos_procedurelist(sosURL)
```

**Arguments**

sosURL            A list. An SOS endpoint (e.g. <http://getit.lteritalia.it/observations/sos/kvp?>).

**Value**

The output of the function is a list with the name and URI (Uniform Resource Identifier) of each procedure.

**Author(s)**

Alessandro Oggioni, PhD (2020) <oggioni.a@irea.cnr.it>

Paolo Tagliolato, PhD (2021) <tagliolato.p@irea.cnr.it>

**References**

Wickham H, Hester J, Ooms J (2021). *xml2: Parse XML*. R package version 1.3.3, <https://CRAN.R-project.org/package=xml2>.

Ooms J (2021). *xslt: Extensible Style-Sheet Language Transformations*. R package version 1.4.3, <https://CRAN.R-project.org/package=xslt>.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.

**Examples**

```
## Not run:
get_sos_procedurelist(
  sosURL = "http://getit.lteritalia.it/observations/service"
)

## End(Not run)
```

---

get\_sos\_procedure\_info

*Obtain the information from a sensor.*

---

**Description**

**[Experimental]** This function obtains the information (metadata) shared by procedure/sensor through Sensor Observation Service (SOS).

**Usage**

```
get_sos_procedure_info(sosURL, procedure)
```

**Arguments**

`sosURL`            A character. The endpoint of the Sensor Observation Service (SOS) service.  
`procedure`        A character. It is a procedure/sensor ID.

**Value**

The output of the function is a character containing attributes such as name of the sensor, id and their description. All the information are collected by requests to Sensor Observation Service (SOS).

**Author(s)**

Alessandro Oggioni, PhD <oggioni.a@irea.cnr.it>  
 Paolo Tagliolato, PhD <tagliolato.p@irea.cnr.it>

**Examples**

```
FP <- get_sos_procedure_info(
  sosURL = "http://getit.lteritalia.it/observations/service",
  procedure = "http://www.get-it.it/sensors/getit.lteritalia.it/procedure/noOwnerDeclared/noModelDeclared/noSer
)
FP

# EURAC (https://www.eurac.edu/)
# eurac_monalisa <- get_sos_procedure_info(
#   sosURL = "http://monalisasos.eurac.edu/sos/service",
#   procedure = "QuantumSensor_nemef2000"
# )
# eurac_monalisa

# obsProsAir <- get_sos_procedure_info(
#   sosURL = "http://getit.lteritalia.it/observations/service",
#   procedure = "http://www.get-it.it/sensors/getit.lteritalia.it/procedure/noOwnerDeclared/noModelDeclared/noSe
# )
# obsProsAir

# NIVA (https://niwa.co.nz)
# NIVA <- get_sos_procedure_info(
#   sosURL = "https://hydro-sos.niwa.co.nz/",
#   procedure = "Water_Quality_Site"
# )
# NIVA
```

---

get_zenodo_data	<i>Obtain the data from a dataset deposited in Zenodo record.</i>
-----------------	---

---

## Description

**[Experimental]** The function returns the dataset, or file(s), deposited in Zenodo record.

## Usage

```
get_zenodo_data(doi, rdata_exist = TRUE)
```

## Arguments

doi	A character. It is the DOI of the Zenodo record.
rdata_exist	A logical. Is the .RData or .rds file in the record we are questioning? Default TRUE.

## Value

a file(s) containing in the Zenodo record.

## Author(s)

Alessandro Oggioni, PhD <oggioni.a@irea.cnr.it>

## Examples

```
## Not run:  
## Not run:  
  
record <- get_zenodo_data(  
  doi = "10.5281/zenodo.7041152", # test dataset  
  rdata_exist = TRUE  
)  
  
## End (Not run)  
  
## End(Not run)
```

---

map_occ_gbif2elter	<i>Harmonize outputs of get_site_speciesOccurrence and map them into eLTER reporting format</i>
--------------------	---

---

## Description

[Experimental]

## Usage

```
map_occ_gbif2elter(x, deimsid)
```

## Arguments

x	A tibble like one that can be obtained by <code>as_tibble(get_site_speciesOccurrence(deimsid, "gbif"))\$gbif</code>
deimsid	A character. The DEIMS.iD of the site from DEIMS-SDR website. DEIMS ID information

## Value

list with the following named elements:

- deimsid: the same deimsid passed in input
- source: one of "gbif", "inat", "obis"
- data\_mapping: tibble structured according to data\_mapping of eLTER reporting format
- reference\_TAXA: tibble structured according to reference\_TAXA of eLTER reporting format
- reference\_VARIABLES: tibble structured according to reference\_VARIABLES of eLTER reporting format

## Author(s)

Paolo Tagliolato, PhD <tagliolato.p@irea.cnr.it>

Martina Zilioli <zilioli.m@irea.cnr.it>

Alessandro Oggioni, PhD <oggioni.a@irea.cnr.it>



---

map_occ_inat2elter	<i>Harmonize outputs of get_site_speciesOccurrence and map them into eLTER reporting format</i>
--------------------	---

---

## Description

[Experimental]

## Usage

```
map_occ_inat2elter(x, deimsid)
```

## Arguments

x	A tibble like one that can be obtained by <code>as_tibble(get_site_speciesOccurrence(deimsid, "gbif"))\$gbif</code>
deimsid	A character. The DEIMS.iD of the site from DEIMS-SDR website. DEIMS ID information

## Value

list with the following named elements:

- deimsid: the same deimsid passed in input
- source: one of "gbif", "inat", "obis"
- data\_mapping: tibble structured according to data\_mapping of eLTER reporting format
- reference\_TAXA: tibble structured according to reference\_TAXA of eLTER reporting format
- reference\_VARIABLES: tibble structured according to reference\_VARIABLES of eLTER reporting format

## Author(s)

Paolo Tagliolato, PhD <tagliolato.p@irea.cnr.it>

Martina Zilioli <zilioli.m@irea.cnr.it>

Alessandro Oggioni, PhD <oggioni.a@irea.cnr.it>

---

map_occ_obis2elter	<i>Harmonize outputs of get_site_speciesOccurrence and map them into eLTER reporting format</i>
--------------------	---

---

## Description

**[Experimental]**

## Usage

```
map_occ_obis2elter(x, deimsid)
```

## Arguments

x	A tibble like one that can be obtained by <code>as_tibble(get_site_speciesOccurrence(deimsid, "gbif"))\$gbif</code>
deimsid	A character. The DEIMS.iD of the site from DEIMS-SDR website. DEIMS ID information

## Value

list with the following named elements:

- deimsid: the same deimsid passed in input
- source: one of "gbif", "inat", "obis"
- data\_mapping: tibble structured according to data\_mapping of eLTER reporting format
- reference\_TAXA: tibble structured according to reference\_TAXA of eLTER reporting format
- reference\_VARIABLES: tibble structured according to reference\_VARIABLES of eLTER reporting format

## Author(s)

Paolo Tagliolato, PhD <tagliolato.p@irea.cnr.it>

Martina Zilioli <zilioli.m@irea.cnr.it>

Alessandro Oggioni, PhD <oggioni.a@irea.cnr.it>

---

package_settings	<i>Package settings that can be changed by the user</i>
------------------	---

---

**Description**

Package settings that can be changed by the user

**Usage**

```
package_settings
```

**Format**

An object of class environment of length 1.

**See Also**

Other package\_customizable\_settings: [get\\_deims\\_base\\_url\(\)](#)

---

plot_agg_map	<i>Map of aggregated time series of MODIS images</i>
--------------	--

---

**Description**

Prepare, show and save an aggregated map of acquired MODIS products. For plotting, the function requires `packageVersion("leaflet") > 2.1.1`

**Usage**

```
plot_agg_map(product, output_dir, site_name, agg_function = "mean")
```

**Arguments**

product	character one of "LST" or "VI"
output_dir	character, where MODIS images were saved This directory is returned by <code>get_site_MODIS()</code> The final map, as png image file will be saved here also.
site_name	character the site (passed from <code>get_site_MODIS()</code> )
agg_function	character either FALSE (the default) or one of "mean", "max", "min". All maps in time series will be aggregated using this function.

**Details**

Read all time series images (from \*.vrt file) in `output_dir` Prepare an aggregation raster of all maps in the time series Save and show a plot of the aggregated map

This function is not exported. It is called by `get_site_MODIS()`

**Value**

Full paths to saved Geotiff rasters

**Author(s)**

Micha Silver, PhD (2020) <silverm@post.bgu.ac.il>

Alessandro Oggioni, PhD (2020) <oggioni.a@irea.cnr.it>

**References**

Pebesma E (2018). "Simple Features for R: Standardized Support for Spatial Vector Data." *The R Journal*, **10**(1), 439–446. doi:10.32614/RJ2018009.

Hijmans RJ (2022). *terra: Spatial Data Analysis*. R package version 1.5-21, <https://CRAN.R-project.org/package=terra>.

---

plot\_timeseries      *Plot a time series of averaged pixel values from MODIS images.*

---

**Description**

Create a time series of averaged pixel values from MODIS images cropped to site boundaries. Display a line plot and save to png. [here](#).

**Usage**

```
plot_timeseries(deimsid, product, output_dir, output_proj = "3035")
```

**Arguments**

deimsid	A character. The DEIMS ID of the site from DEIMS-SDR website. DEIMS ID information <a href="#">here</a> .
product	A character. The requested product. One of: "LST", "VI". Default is "VI".
output_dir	a character, where MODIS images were saved This directory is returned by <code>get_site_MODIS()</code> The final graph as png image file will be saved here also.
output_proj	character: The EPSG code of output rasters

**Details**

Read all images in `output_dir` and prepare line plots of average pixel values over the site boundary for each band.

This function is not exported. It is called by `get_site_MODIS()`

**Value**

Full path to the saved png image.

**Author(s)**

Micha Silver, PhD (2020) <silverm@post.bgu.ac.il>

Alessandro Oggioni, PhD (2020) <oggioni.a@irea.cnr.it>

**References**

Pebesma E (2018). "Simple Features for R: Standardized Support for Spatial Vector Data." *The R Journal*, **10**(1), 439–446. doi:10.32614/RJ2018009.

Hijmans RJ (2022). *terra: Spatial Data Analysis*. R package version 1.5-21, <https://CRAN.R-project.org/package=terra>.

**Examples**

```
## Not run:
# Example in Northern Negev LTER
deimsid <- "https://deims.org/871a90b2-e372-456a-93e3-518ad1e11239"
output_dir <- tempdir()
png_files <- ReLTER::plot_timeseries(deimsid, product = "VI",
  output_dir = output_dir, output_proj = "6991")
message("Output plots: ", png_files)

## End(Not run)
```

---

produce\_network\_points\_map

*Provide a map (image) of sites in an eLTER Network.*

---

**Description**

**[Stable]** Return a image map object of all of the eLTER sites belonging to an eLTER Network (e.g. [LTER Italy network](#)), as a stored into [DEIMS-SDR](#).

**Usage**

```
produce_network_points_map(networkDEIMSID, countryCode)
```

**Arguments**

**networkDEIMSID** A character. The DEIMS ID of the network from DEIMS-SDR website. DEIMS ID information [here](#) and Complete list of ILTER networks [here](#).

**countryCode** A character following the ISO 3166-1 alpha-3 codes. This ISO convention consists of three-letter country codes as defined in ISO 3166-1. The ISO 3166 standard published by the International Organization for Standardization (ISO), to represent countries, dependent territories, and special areas of geographical interest. The map produced by this function will be limited only to the country indicated in this parameter, if the network has a extraterritorial sites those will not represented.

**Value**

The output of the function is a tmap plot containing an image of geographic distribution of the network of sites present in the chosen country.

**The function output****Author(s)**

Alessandro Oggioni, PhD (2020) <oggioni.a@irea.cnr.it>

**References**

- Ooms J (2014). “The jsonlite Package: A Practical and Consistent Mapping Between JSON Data and R Objects.” *arXiv:1403.2805 [stat.CO]*. <https://arxiv.org/abs/1403.2805>.
- Pebesma E (2018). “Simple Features for R: Standardized Support for Spatial Vector Data.” *The R Journal*, **10**(1), 439–446. doi:10.32614/RJ2018009.
- Tennekes M (2018). “tmap: Thematic Maps in R.” *Journal of Statistical Software*, **84**(6), 1–39. doi:10.18637/jss.v084.i06.
- Wickham H, François R, Henry L, Müller K (2022). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.9, <https://CRAN.R-project.org/package=dplyr>.
- Müller K, Wickham H (2022). *tibble: Simple Data Frames*. R package version 3.1.7, <https://CRAN.R-project.org/package=tibble>.
- Wickham H (2022). *httr: Tools for Working with URLs and HTTP*. R package version 1.4.3, <https://CRAN.R-project.org/package=httr>.

**Examples**

```
## Not run:  
# Italian sites  
map <- produce_network_points_map(  

```

```

networkDEIMSID =
  "https://deims.org/network/7fef6b73-e5cb-4cd2-b438-ed32eb1504b3",
  countryCode = "ITA"
)
map

# German sites
map_LTERGermanSites <- produce_network_points_map(
  networkDEIMSID =
    "https://deims.org/networks/e904354a-f3a0-40ce-a9b5-61741f66c824",
    countryCode = "DEU"
)
map_LTERGermanSites +
  tmap::tm_compass(type = "8star", position = c("right", "bottom")) +
  tmap::tm_scale_bar(position = c("right", "bottom"))

## End(Not run)

```

---

produce\_site\_map

*Provide a map object of a sites LTER.*

---

## Description

**[Stable]** This function produces a map of the site boundaries as provided by the [DEIMS-SDR catalogue](#), within a given country and network.

## Usage

```

produce_site_map(
  deimsid,
  countryCode,
  listOfSites,
  gridNx,
  gridNy,
  width = 0.25,
  height = 0.25,
  bboxXMin = 0,
  bboxXMax = 0,
  bboxYMin = 0,
  bboxYMax = 0,
  show_map = FALSE
)

```

## Arguments

**deimsid** A character. The DEIMS ID of network from DEIMS-SDR website. DEIMS ID information [here](#).

<code>countryCode</code>	A character following the SO 3166-1 alpha-3 codes. This ISO convention consists of three-letter country codes as defined in ISO 3166-1, part of the ISO 3166 standard published by the International Organization for Standardization (ISO), to represent countries, dependent territories, and special areas of geographical interest. The map produced by this function will be limited to the country indicated in this parameter; if the network has a extraterritorial sites those will not represented.
<code>listOfSites</code>	A sf. List of sites of specific network. This list is needed for showing another points on the map.
<code>gridNx</code>	A double. A numeric vector or unit object specifying x-location of viewports about country provided by <code>countryCode</code> parameter.
<code>gridNy</code>	A double. A numeric vector or unit object specifying y-location of viewports about country provided by <code>countryCode</code> parameter.
<code>width</code>	A double. A numeric vector or unit object specifying width of viewports about country provided by <code>countryCode</code> parameter. Default 0.25.
<code>height</code>	A double. A numeric vector or unit object specifying height of viewports about country provided by <code>countryCode</code> parameter. Default 0.25.
<code>bboxXMin</code>	A double. A numeric for add some unit of a bbox provided by centroid of the site. Default 0.
<code>bboxXMax</code>	A double. A numeric for add some unit of a bbox provided by centroid of the site. Default 0.
<code>bboxYMin</code>	A double. A numeric for add some unit of a bbox provided by centroid of the site. Default 0.
<code>bboxYMax</code>	A double. A numeric for add some unit of a bbox provided by centroid of the site. Default 0.
<code>show_map</code>	A boolean. When TRUE the immagine of map will be plotted. Default FALSE.

### **Value**

The output of the function is an image of the boundary of the site, OSM as base map and all country sites map.



## The function output



## Author(s)

Alessandro Oggioni, PhD (2020) <oggioni.a@irea.cnr.it>

## References

- Pebesma E (2018). “Simple Features for R: Standardized Support for Spatial Vector Data.” *The R Journal*, **10**(1), 439–446. doi:10.32614/RJ2018009.
- Ooms J (2014). “The jsonlite Package: A Practical and Consistent Mapping Between JSON Data and R Objects.” *arXiv:1403.2805 [stat.CO]*. <https://arxiv.org/abs/1403.2805>.
- Müller K, Wickham H (2022). *tibble: Simple Data Frames*. R package version 3.1.7, <https://CRAN.R-project.org/package=tibble>.
- Hijmans RJ (2022). *raster: Geographic Data Analysis and Modeling*. R package version 3.5-15, <https://CRAN.R-project.org/package=raster>.
- Bivand R, Rundel C (2021). *rgeos: Interface to Geometry Engine - Open Source ('GEOS')*. R package version 0.5-9, <https://CRAN.R-project.org/package=rgeos>.
- Dunnington D (2019). *rosm: Plot Raster Map Tiles from Open Street Map and Other Sources*. R package version 0.2.5, <https://CRAN.R-project.org/package=rosm>.
- Tennekes M (2018). “tmap: Thematic Maps in R.” *Journal of Statistical Software*, **84**(6), 1–39. doi:10.18637/jss.v084.i06.
- R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.

## Examples

```
## Not run:
# Example of Lange Bramke site
sitesNetwork <- get_network_sites(
```

```

networkDEIMSID =
  "https://deims.org/networks/e904354a-f3a0-40ce-a9b5-61741f66c824"
)
map <- produce_site_map(
  deimsid = "https://deims.org/8e24d4f8-d6f6-4463-83e9-73cac2fd3f38",
  countryCode = "DEU",
  listOfSites = sitesNetwork,
  gridNx = 0.2,
  gridNy = 0.7
)

# Example of Eisenwurzten site
sitesNetwork <- get_network_sites(
  networkDEIMSID =
    "https://deims.org/networks/d45c2690-dbef-4dbc-a742-26ea846edf28"
)
map <- produce_site_map(
  deimsid = "https://deims.org/d0a8da18-0881-4ebe-bccf-bc4cb4e25701",
  countryCode = "AUT",
  listOfSites = sitesNetwork,
  gridNx = 0.2,
  gridNy = 0.7
)

# Example of Lake Maggiore site
sitesNetwork <- get_network_sites(
  networkDEIMSID =
    "https://deims.org/network/7fef6b73-e5cb-4cd2-b438-ed32eb1504b3"
)
# In the case of Italian sites are selected only true sites and excluded the
# macrosites.
sitesNetwork <- (sitesNetwork[!grepl('^IT', sitesNetwork$title),])
sf::st_crs(sitesNetwork) = 4326
siteMap <- produce_site_map(
  deimsid = "https://deims.org/f30007c4-8a6e-4f11-ab87-569db54638fe",
  countryCode = "ITA",
  listOfSites = sitesNetwork,
  gridNx = 0.7,
  gridNy = 0.35,
  show_map = TRUE
)
siteMap

# with show_map = FALSE
siteMap <- produce_site_map(
  deimsid = "https://deims.org/f30007c4-8a6e-4f11-ab87-569db54638fe",
  countryCode = "ITA",
  listOfSites = sitesNetwork,
  gridNx = 0.7,
  gridNy = 0.35
)
siteMap

```

```
## End(Not run)
```

---

```
produce_site_observedProperties_pie
```

*Produce a pie chart of the observed properties collected in a site LTER.*

---

## Description

**[Stable]** Return a pie chart of Environmental observed properties, as a stored in [DEIMS-SDR catalogue](#), of a single eLTER site.

## Usage

```
produce_site_observedProperties_pie(deimsid)
```

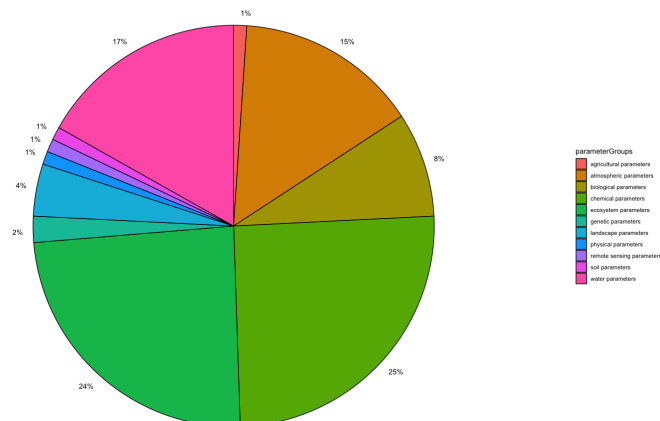
## Arguments

`deimsid` A character. It is the DEIMS ID of site/network from DEIMS-SDR website. DEIMS ID information [here](#).

## Value

The output of the function is a pie chart and a tibble. The percentages, as a label in the pie charts and in the output table ( column 'perc'), refer to the number of the observed properties, belonging to a type (e.g. biological, atmospheric, etc.), measured compared to all of observed properties measured into selected eLTER site. This function allows to show what type of observed properties are most measured into a site. In the example below the atmospheric observed properties corresponds to the 15 percent of all observed properties measured into the site.

## The function output



**Author(s)**

Alessandro Oggioni, PhD (2020) <oggioni.a@irea.cnr.it>

**References**

Müller K, Wickham H (2022). *tibble: Simple Data Frames*. R package version 3.1.7, <https://CRAN.R-project.org/package=tibble>.

Wickham H, François R, Henry L, Müller K (2022). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.9, <https://CRAN.R-project.org/package=dplyr>.

Neuwirth E (2022). *RColorBrewer: ColorBrewer Palettes*. R package version 1.1-3, <https://CRAN.R-project.org/package=RColorBrewer>.

Wickham H (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. ISBN 978-3-319-24277-4, <https://ggplot2.tidyverse.org>.

Pedersen TL (2021). *ggforce: Accelerating 'ggplot2'*. R package version 0.3.3, <https://CRAN.R-project.org/package=ggforce>.

**Examples**

```
## Not run:
pie <- produce_site_observedProperties_pie(
  deimsid = "https://deims.org/f30007c4-8a6e-4f11-ab87-569db54638fe"
)
pie

## End(Not run)
```

---

```
produce_site_observedProperties_waffle
```

*Produce a waffle chart of the observed properties collected in a site LTER.*

---

**Description**

**[Stable]** Return a waffle chart of Environmental observed properties, as a stored in [DEIMS-SDR catalogue](#), of a single eLTER site.

**Usage**

```
produce_site_observedProperties_waffle(deimsid)
```

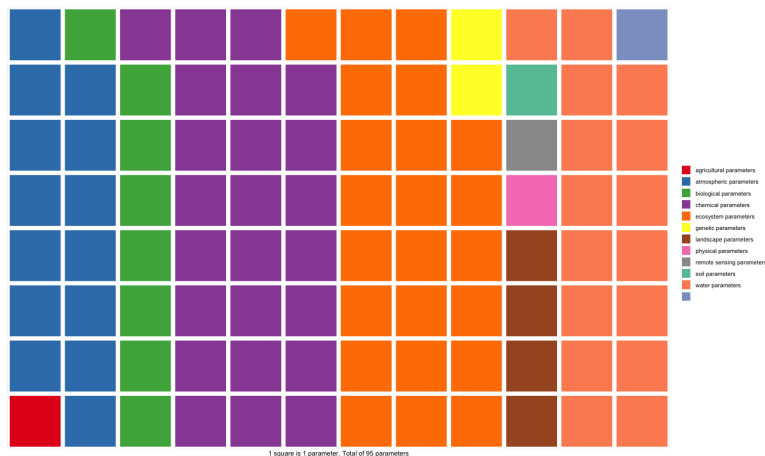
**Arguments**

deimsid            A character. The DEIMS ID of site/network from: DEIMS-SDR website. DEIMS ID information [here](#).

## Value

The output of the function is a waffle chart and a tibble. Each of the squares represents a observed properties measured into the selected eLTER site. The observed properties with the same color belong to the same group (e.g. biological, atmospheric, etc.).

## The function output



## Author(s)

Alessandro Oggioni, PhD (2020) <oggioni.a@irea.cnr.it>

## References

Müller K, Wickham H (2022). *tibble: Simple Data Frames*. R package version 3.1.7, <https://CRAN.R-project.org/package=tibble>.

Wickham H, François R, Henry L, Müller K (2022). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.9, <https://CRAN.R-project.org/package=dplyr>.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.

Neuwirth E (2022). *RColorBrewer: ColorBrewer Palettes*. R package version 1.1-3, <https://CRAN.R-project.org/package=RColorBrewer>.

Rudis B, Gandy D (2017). *waffle: Create Waffle Chart Visualizations in R*. R package version 0.7.0, <https://CRAN.R-project.org/package=waffle>.

## Examples

```
## Not run:
waffle <- produce_site_observedProperties_waffle(
  deimsid = "https://deims.org/f30007c4-8a6e-4f11-ab87-569db54638fe"
)
```

```
waffle  
## End(Not run)
```

---

produce\_site\_qrcode    *Obtain the QRCode of any DEIMS-SDR entities.*

---

### Description

**[Stable]** Return a QR code image of any provided DEIMS ID (e.g. dataset, site, activity).

### Usage

```
produce_site_qrcode(deimsid, do_plot = FALSE)
```

### Arguments

deimsid	A character. The DEIMS ID of entities from DEIMS-SDR website. DEIMS ID information <a href="#">here</a> .
do_plot	A boolean. Plot the computed QRCode. Default FALSE.

### Value

The QR code as a logical matrix with "qr\_code" class.

### The function output



### Author(s)

Alessandro Oggioni, PhD (2020) <oggioni.a@irea.cnr.it>

## References

Teh V, Onkelinx T (2021). *qrcode: Generate QRcodes with R. Version 0.1.4*. doi:10.5281/zenodo.5040088, <https://thierryo.github.io/qrcode/>.

## Examples

```
qrcode <- produce_site_qrcode(  
  deimsid = "https://deims.org/f30007c4-8a6e-4f11-ab87-569db54638fe"  
)  
plot(qrcode)  
  
a <- produce_site_qrcode(  
  deimsid = "https://deims.org/f30007c4-8a6e-4f11-ab87-569db54638fe",  
  do_plot = TRUE  
)
```

---

produce\_zenodo\_record *Create a record in Zenodo*

---

## Description

**[Experimental]** This function allows to deposit a record to **Zenodo** repository. The function use the functions implemented by zen4r package. Blondel, Emmanuel, & Barde, Julien. (2021). zen4R: R Interface to Zenodo REST API (0.5-2). Zenodo. <https://doi.org/10.5281/zenodo.5741143>.

## Usage

```
produce_zenodo_record(  
  mytoken,  
  myfiles,  
  delim,  
  upload_rdata = FALSE,  
  record_type = "dataset",  
  record_title,  
  record_description,  
  record_authors,  
  record_license = "CC-BY-SA-4.0",  
  record_accessRight = "open",  
  record_version = "1.0",  
  record_language = "eng",  
  record_keywords,  
  record_relatedIdentifier,  
  record_communities = c("lter-italy", "elter"),  
  record_grants,  
  record_publish = FALSE  
)
```

**Arguments**

mytoken	A character. Scopes assign permissions to your personal access token. A personal access token works just like a normal OAuth access token for authentication against the API. This token can be created at <a href="#">application page</a> .
myfiles	A character. The list of the file(s) to deposit in Zenodo.
delim	A character. Provide the character used to separate fields within a record. Only if you want to upload to Zenodo a dataset (see param record_type below) with extension 'csv', 'tsv' or 'txt' and if you want to upload also a copy of your dataset in Rdata format (see param upload_rdata below).
upload_rdata	A logical. Do you want also upload a version of the dataset in RData format onto this record? Default FALSE.
record_type	A character. A type of record among the following values: 'publication', 'poster', 'presentation', 'dataset', 'image', 'video', 'software', 'lesson', 'physicalobject', 'other'. Default 'dataset'.
record_title	A character. The title of the record.
record_description	A character. The description of the record.
record_authors	A tibble. It is a list of creator(s) of the record. The approach is to use the firstname, lastname, affiliation, orcid of the authors of the record. Please follow the example.
record_license	A character. It is the license with which the record is released. The license should be set with the Zenodo id of the license. Default "CC-BY-SA-4.0". The supported licenses values are from <a href="https://opendefinition.org">https://opendefinition.org</a> and <a href="https://spdx.org">https://spdx.org</a> .
record_accessRight	A character. Default "open". Other options are: 'embargoed', 'restricted' and 'closed'.
record_version	A character. It is a version of the record. Default "1.0".
record_language	A character. It is the language of the record. Only one value is possible. Default "eng".
record_keywords	A character. A multiple values are possible. The keyword to the record of record.
record_relatedIdentifier	A tibble. It is the related entities (e.g. scientific paper, data paper, etc.) with the dataset. The tibble is composed by 2 variables: relation and identifier. Relation can be one of among following values: "isCitedBy", "cites", "isSupplementTo", "isSupplementedBy", "isNewVersionOf", "isPreviousVersionOf", "isPartOf", "hasPart", "compiles", "isCompiledBy", "isIdenticalTo", "isAlternateIdentifier". While identifier is any type of identifier (e.g. DOI).
record_communities	A character. It is a name of communities as created in Zenodo. A multiple values are possible. Default "lter-italy" and "elter".



- record\_grants A character. A multiple values are possible. Put a list of project identifier as well as showed by the European Commission via [OpenAIRE](#).
- record\_publish A logical. State whether the Zenodo record is to be published. The parameter publish can be set to TRUE (to use CAUTIOUSLY, only if you want to publish your record). Default FALSE.

### Value

A link (URL) of the deposited record in Zenodo. The user must then:

1. visit the webpage of record, 2. check the information provided, 3. 'save' and 'publish' the record on Zenodo repository, 4. use the DOI to cite the record.

### Author(s)

Alessandro Oggioni, PhD <oggioni.a@irea.cnr.it>

### See Also

zen4R documentation <https://github.com/eblondel/zen4R/wiki>

### Examples

```
## Not run:
## Not run:

authors <- tibble::tibble(
  name = c("Luke", "Leia"),
  surname = c("Skywalker", "Organa"),
  affiliation = c("Tatooine", "Alderaan"),
  orcid = c("0000-0002-7997-219X", "0000-0002-7997-219X")
)
keywords <- c("Star Wars", "species", "films", "planets")
relatedIdentifiers <- tibble::tibble(
  relation = c("isSupplementTo", "isPartOf"),
  identifier = c("10.1038/s4150-01-0032", "10.1016/j.2051.06.026")
)
grants <- c("871128", "654359", "871126")
produce_zenodo_record(
  mytoken = mytoken, # generate your Zenodo token
  myfiles = myfile, # provide your file(s)
  delim = ";",
  upload_rdata = TRUE,
  record_title = "The title",
  record_description = "This is the description of the record.",
  record_authors = authors,
  record_keywords = keywords,
  record_relatedIdentifier = relatedIdentifiers,
  record_communities = "lter-italy",
  record_grants = grants,
  record_publish = FALSE
)
```

```
## End(Not run)

## End (Not run)
```

---

```
produce_zenodo_record_from_elter_reporting
  Create a record in Zenodo from eLTER data reporting format
```

---

### Description

**[Experimental]** This function allows to deposit a record to **Zenodo** repository with the eLTER data reporting format. The function use the functions implemented by zen4r package. Blondel, Emmanuel, & Barde, Julien. (2021). zen4R: R Interface to Zenodo REST API (0.5-2). Zenodo. <https://doi.org/10.5281/zenodo.5741143>.

### Usage

```
produce_zenodo_record_from_elter_reporting(
  x,
  saveRDS = FALSE,
  filepath = tempdir(),
  filename,
  mytoken,
  record_title,
  record_description,
  record_authors,
  record_license = "CC-BY-SA-4.0",
  record_accessRight = "open",
  record_version = "1.0",
  record_language = "eng",
  record_keywords,
  record_relatedIdentifier,
  record_communities = c("lter-italy", "elter"),
  record_grants
)
```

### Arguments

x	A list like the one created by function reporting_produce_data_object_v1.3
saveRDS	A logical. Save also object in RDS format. Defaults to FALSE.
filepath	A character file path. Defaults to temporary directory
filename	A character. Optional filename associated with the object, of the form provided as output by the function reporting_compose_file_name. Defaults to random string

mytoken	A character. Scopes assign permissions to your personal access token. A personal access token works just like a normal OAuth access token for authentication against the API. This token can be created at application <a href="#">page</a> .
record_title	A character. The title of the record.
record_description	A character. The description of the record.
record_authors	A tibble. It is a list of creator(s) of the record. The approach is to use the firstname, lastname, affiliation, orcid of the authors of the record. Please follow the example.
record_license	A character. It is the license with which the record is released. The license should be set with the Zenodo id of the license. Default "CC-BY-SA-4.0". The supported licenses values are from <a href="https://opendefinition.org">https://opendefinition.org</a> and <a href="https://spdx.org">https://spdx.org</a> .
record_accessRight	A character. Default "open". Other options are: 'embargoed', 'restricted' and 'closed'.
record_version	A character. It is a version of the record. Default "1.0".
record_language	A character. It is the language of the record. Only one value is possible. Default "eng".
record_keywords	A character. A multiple values are possible. The keyword to the record of record.
record_relatedIdentifier	A tibble. It is the related entities (e.g. scientific paper, data paper, etc.) with the dataset. The tibble is composed by 2 variables: relation and identifier. Relation can be one of among following values: "isCitedBy", "cites", "isSupplementTo", "isSupplementedBy", "isNewVersionOf", "isPreviousVersionOf", "isPartOf", "hasPart", "compiles", "isCompiledBy", "isIdenticalTo", "isAlternateIdentifier". While identifier is any type of identifier (e.g. DOI).
record_communities	A character. It is a name of communities as created in Zenodo. A multiple values are possible. Default "lter-italy" and "elter".
record_grants	A character. A multiple values are possible. Put a list of project identifier as well as showed by the European Commission via <a href="#">OpenAIRE</a> .

### Value

A link (URL) of the deposited record in Zenodo. The user must then:

1. visit the webpage of record,
2. check the information provided,
3. 'save' and 'publish' the record on Zenodo repository,
4. use the DOI to cite the record.

### Author(s)

Alessandro Oggioni, PhD <oggioni.a@irea.cnr.it>

Paolo Tagliolato, PhD <tagliolato.p@irea.cnr.it>

**See Also**

zen4R documentation <https://github.com/eblondel/zen4R/wiki>

**Examples**

```
## Not run:
## Not run:

deimsid <- "https://deims.org/8eda49e9-1f4e-4f3e-b58e-e0bb25dc32a6"
time_span <- 2015 # e.g. whole year
time_span <- "20150302-20180415" # e.g. span between two dates
data_topic <- "VEG" # data provider defined abbreviation of "vegetation"
variable_group <- "SPECCOVER" # data provider defined abbreviation
version <- "V20220907"

filename <- reporting_compose_file_name(
  deimsid = deimsid,
  data_topic = data_topic,
  variable_group = variable_group,
  time_span = time_span,
  version = version
)

data <- dplyr::tribble(
  ~id, ~value,
  1, 7.5,
  2, 4.2
)

station <- dplyr::tribble(
  ~SITE_CODE, ~STATION_CODE, ~STYPE, ~LAT, ~LON, ~ALTITUDE,
  deimsid, "IP2", "AREA", 45.340805, 7.88887495, 265
)

method <- dplyr::tribble(
  ~VARIABLE, ~METH_DESCR,
  "COVE_F", "Analysis of ammonium..."
)

research_object <- reporting_produce_data_object_v1.3(
  filename = filename,
  deimsid = deimsid,
  data = data,
  station = station,
  method = method
)

authors <- tibble::tibble(
  name = c("Luke", "Leia"),
  surname = c("Skywalker", "Organa"),
  affiliation = c("Tatooine", "Alderaan"),
  orcid = c("0000-0002-7997-219X", "0000-0002-7997-219X")
)

keywords <- c("Star Wars", "species", "films", "planets")
```

```

relatedIdentifiers <- tibble::tibble(
  relation = c("isSupplementTo", "isPartOf"),
  identifier = c("10.1038/s4150-01-0032", "10.1016/j.2051.06.026")
)
grants <- c("871128", "654359", "871126")

produce_zenodo_record_from_elter_reporting(
  x = research_object,
  saveRDS = TRUE,
  filepath = ".",
  filename = filename,
  mytoken = mytoken, # generate your Zenodo token
  record_title = "The title of eLTER reporting",
  record_description = "This is the description of the record of eLTER data
  reporting format.",
  record_authors = authors,
  record_license = "CC-BY-SA-4.0",
  record_accessRight = "open",
  record_version = "1.0",
  record_language = "eng",
  record_keywords = keywords,
  record_relatedIdentifier = relatedIdentifiers,
  record_communities = "lter-italy",
  record_grants = grants
)

## End(Not run)

## End (Not run)

```

---

```

reporting_compose_file_name
eLTER reporting format naming convention for files

```

---

## Description

**[Experimental]** Compose file name following eLTER naming convention

## Usage

```

reporting_compose_file_name(
  deimsid = NULL,
  country_code = NULL,
  site_name = NULL,
  data_topic,
  variable_group = "",
  time_span,
  version = Sys.Date() %>% format("V%Y%m%d")
)

```

**Arguments**

deimsid	A character The DEIMS ID of the site from DEIMS-SDR website. More information about DEIMS ID in this pages: <a href="#">page</a> .
country_code	A character automatically evaluated if DEIMS ID is provided. Otherwise reference to the country of the site as two-digit country code according to ISO 3166-1 alpha-2.
site_name	A character Automatically evaluated if DEIMS ID is provided. Otherwise the name of the site according to DEIMS-SDR, if the name is too long the site name can be shortened.
data_topic	A character. Max 5-digit code for data topic or observation programme, e.g. METEO (Meteorology), BIODIV (Biodiversity), DEPO (deposition), GHG (Green House gas), SW (Soil water), VEG (Vegetation). The abbreviation is defined by the data provider depending on the data.
variable_group	A character. Optional, list of variables or variable groups contained in the data. The abbreviation is defined by the data provider depending on the data.
time_span	A numeric or a character. Time span covered in the data.
version	version in format "YYYYMMDD". Data version in the format "V"YYYYMMDD. Defaults to current date.

**Value**

filename (without extension) following naming convention

**Note**

This method must be intended as a signpost for future implementation

**Author(s)**

Paolo Tagliolato, PhD <tagliolato.p@irea.cnr.it>

Alessandro Oggioni, PhD <oggioni.a@irea.cnr.it>

**See Also**

Peterseil, Geiger et al. (2020) Field Specification for data reporting. Technical Document. Tech-Doc.01. EU Horizon 2020 eLTER PLUS Project, Grant agreement No. 871128 <https://zenodo.org/record/6373410>

**Examples**

```
## Not run:
## Not run:

deimsid <- "https://deims.org/8eda49e9-1f4e-4f3e-b58e-e0bb25dc32a6"
time_span <- 2015 # e.g. whole year
# time_span <- "20150302-20180415" # e.g. span between two dates
data_topic <- "VEG" # data provider defined abbreviation of "vegetation"
```

```

variable_group <- "SPECCOVER" # data provider defined abbreviation
version <- "V20220907"

filename <- reporting_compose_file_name(
  deimsid = deimsid,
  data_topic = data_topic,
  variable_group = variable_group,
  time_span = time_span,
  version = version
)

## End (Not run)

## End(Not run)

```

---

reporting\_produce\_data\_object\_v1.3

*Compose an eLTER Data Reporting Format object*

---

## Description

**[Experimental]** Given several tables, creates an eLTER data reporting format object

## Usage

```

reporting_produce_data_object_v1.3(
  data = NULL,
  station = NULL,
  method = NULL,
  reference = NULL,
  event = NULL,
  sample = NULL,
  licence = "",
  deimsid = "",
  data_type = "measurement",
  filename = NULL
)

```

## Arguments

data	A tibble. See eLTER data specification format for details
station	A tibble
method	A tibble
reference	A tibble
event	A tibble
sample	A tibble

licence	A character
deimsid	A character. The DEIMS ID of the site from DEIMS-SDR website. DEIMS ID information <a href="#">here</a> .
data_type	A character. Data must be provided by one of measurement or mapping. Default 'measurement'
filename	optional filename associated with the object, of the form provided as output by the function <code>reporting_compose_file_name</code>

**Value**

list with eLTER reporting format slots

**Note**

This method must be intended as a signpost for future implementation

**Author(s)**

Paolo Tagliolato, PhD <tagliolato.p@irea.cnr.it>

Alessandro Oggioni, PhD <oggioni.a@irea.cnr.it>

**See Also**

Peterseil, Geiger et al. (2020) Field Specification for data reporting. Technical Document. Tech-Doc.01. EU Horizon 2020 eLTER PLUS Project, Grant agreement No. 871128 <https://zenodo.org/record/6373410>

**Examples**

```
## Not run:
## Not run:

deimsid <- "https://deims.org/8eda49e9-1f4e-4f3e-b58e-e0bb25dc32a6"
time_span <- 2015 # e.g. whole year
# time_span <- "20150302-20180415" # e.g. span between two dates
data_topic <- "VEG" # data provider defined abbreviation of "vegetation"
variable_group <- "SPECCOVER" # data provider defined abbreviation
version <- "V20220907"

filename <- reporting_compose_file_name(
  deimsid = deimsid,
  data_topic = data_topic,
  variable_group = variable_group,
  time_span = time_span,
  version = version
)

data <- tibble::tribble(
  ~`SITE_CODE`, ~`VARIABLE`, ~`TIME`, ~`VALUE`, ~`UNIT`,
  "https://deims.org/8eda49e9-1f4e-4f3e-b58e-e0bb25dc32a6", "TEMP", "2016-03-15", "5.5", "°C",
```



```

      "https://deims.org/8eda49e9-1f4e-4f3e-b58e-e0bb25dc32a6", "PREC", "2016-03-03", "10.2", "mm",
      "https://deims.org/8eda49e9-1f4e-4f3e-b58e-e0bb25dc32a6", "TEMP", "2016-02-15", "2.5", "°C",
      "https://deims.org/8eda49e9-1f4e-4f3e-b58e-e0bb25dc32a6", "NH4N", "2016-03", "5.5", "mg/l",
      "https://deims.org/8eda49e9-1f4e-4f3e-b58e-e0bb25dc32a6", "SO4S", "2016-03", "10.2", "mg/l",
      "https://deims.org/8eda49e9-1f4e-4f3e-b58e-e0bb25dc32a6", "CA", "2016-03", "2.5", "mg/l"
    )
  station <- dplyr::tribble(
    ~SITE_CODE, ~STATION_CODE, ~STYPE, ~LAT, ~LON, ~ALTITUDE,
    deimsid, "IP2", "AREA", 45.340805, 7.88887495, 265
  )
  method <- dplyr::tribble(
    ~VARIABLE, ~METH_DESCR,
    "COVE_F", "Analysis of ammonium..."
  )

  research_object <- reporting_produce_data_object_v1.3(
    filename = filename,
    deimsid = deimsid,
    data = data,
    station = station,
    method = method
  )

  ## End(Not run)
  ## End (Not run)

```

---

reporting\_save\_archive

*Creates an archive with files following the eLTER reportingFormat*

---

## Description

**[Experimental]** Creates a zip archive "filename".zip

## Usage

```

reporting_save_archive(
  x,
  filename = NULL,
  filepath = tempdir(),
  saveRDS = FALSE
)

```

## Arguments

x                    A list like the one created by function reporting\_produce\_data\_object\_v1.3

filename	A character. Optional filename associated with the object, of the form provided as output by the function <code>reporting_compose_file_name</code> . Defaults to random string
filepath	A character file path. Defaults to temporary directory
saveRDS	A logical. Save also object in RDS format. Defaults to FALSE

**Value**

named A list containing paths to saved files filepaths. Slots are named "zip" and possibly "RDS".

**Note**

This method must be intended as a signpost for future implementation

**Author(s)**

Paolo Tagliolato, PhD <tagliolato.p@irea.cnr.it>

Alessandro Oggioni, PhD <oggioni.a@irea.cnr.it>

**See Also**

Peterseil, Geiger et al. (2020) Field Specification for data reporting. Technical Document. Tech-Doc.01. EU Horizon 2020 eLTER PLUS Project, Grant agreement No. 871128 <https://zenodo.org/record/6373410>

**Examples**

```
## Not run:
## Not run:

deimsid <- "https://deims.org/8eda49e9-1f4e-4f3e-b58e-e0bb25dc32a6"
time_span <- 2015 # e.g. whole year
# time_span <- "20150302-20180415" # e.g. span between two dates
data_topic <- "VEG" # data provider defined abbreviation of "vegetation"
variable_group <- "SPECCOVER" # data provider defined abbreviation
version <- "V20220907"

filename <- reporting_compose_file_name(
  deimsid = deimsid,
  data_topic = data_topic,
  variable_group = variable_group,
  time_span = time_span,
  version = version
)

data <- dplyr::tribble(
  ~id, ~value,
  1, 7.5,
  2, 4.2
)

station <- dplyr::tribble(
```

```

    ~SITE_CODE, ~STATION_CODE, ~STYPE, ~LAT,      ~LON,      ~ALTITUDE,
    deimsid,    "IP2",          "AREA",  45.340805, 7.88887495, 265
  )

  method <- dplyr::tribble(
    ~VARIABLE, ~METH_DESCR,
    "COVE_F",  "Analysis of ammonium..."
  )

  research_object <- reporting_produce_data_object_v1.3(
    filename = filename,
    deimsid = deimsid,
    data = data,
    station = station,
    method = method
  )

  archive <- reporting_save_archive(
    x = research_object,
    # obtained from the function `reporting_produce_data_object_v1.3()`
    filename = filename,
    # obtained from the function `reporting_compose_file_name()`
    filepath = ".",
    saveRDS = TRUE
  )

  ## End (Not run)

  ## End(Not run)

```

---

```
save_occ_eLTER_reporting_Archive
```

*Creates an archive with files following the eLTER reportingFormat*

---

## Description

**[Experimental]** Creates a zip archive named `biodiv_occurrence_site_"deimsid_code"_"source".zip` where `"deimsid_code"` is the uuid in the last part of the `deimsid`, and `"source"` is one of `"gbif"`, `"inat"`, `"obis"`

## Usage

```
save_occ_eLTER_reporting_Archive(lterReportOut, path = tempdir())
```

## Arguments

`lterReportOut` A list like the one created by `map_occ_gbif2elter`  
`path` path of the zip file. Defaults to temporary folder

**Value**

the path to the created file

**Author(s)**

Paolo Tagliolato, PhD <tagliolato.p@irea.cnr.it>

---

set_deims_base_url	<i>Set DEIMS-SDR API base URL</i>
--------------------	-----------------------------------

---

**Description**

Set DEIMS-SDR API base URL

**Usage**

```
set_deims_base_url(url = "https://deims.org/", force = FALSE)
```

**Arguments**

url	A character. Set the base URL to DEIMS-SDR.
force	A boolean. Default FALSE.

---

taxon_id_pesi	<i>Provide a taxon ID <math>R</math>href<a href="https://en.wikipedia.org/wiki/LSID">https://en.wikipedia.org/wiki/LSID</a> to a taxon list.</i>
---------------	--

---

**Description**

**[Stable]** This function provide a taxon ID, usually a **LSID**, from a taxonomic list. The input of the function is a csv file with a list of taxa. The Taxon ID provided by this function is currently taken from Pan-European Species directories Infrastructure - **PESI**. This function takes advantage of taxize's eubon\_search function <https://docs.ropensci.org/taxize/> and the **PESI RestAPI**.

**Usage**

```
taxon_id_pesi(table, taxaColumn)
```

**Arguments**

table	A data.frame containing column with a taxa (e.g. Sphaerosoma seidlitzii, Malthinus, etc.).
taxaColumn	A numeric that identify the column containing taxa value.

**Value**

The output of the function is a tibble containing all the columns provided as input and new columns as: 'canonicalName', 'authorship', 'synonyms', 'LSID', 'url', 'accordingTo', 'checkStatus' gathered from PESI.

An example to export dataset obtained by this function is: `datasetMerged <- dplyr::bind_rows(table)`  
`write.csv( datasetMerged, "table.csv", row.names = FALSE, fileEncoding = "UTF-8" )`

Someone could have problems of characters encoding when CSV file is written. To resolve we suggest two different solutions:

Solution 1 -

1. Open the CSV in Notepad.
2. Click "File" and "Save As".
3. In the new popup that displays, select "ANSI" from the "Encoding" field.
4. Click "Save".
5. Now, you should be able to open the file in Excel and display the characters correctly.

Solution 2 -

1. Open Excel
2. Click "File" and "New"
3. Click on the "Data" tab
4. Click "From Text" and select the CSV file
5. Select "Delimited"
6. For "File origin", select "65001 : Unicode (UTF-8)"
7. Click "Next"
8. Select "Comma"
9. Click "Finish"
10. Excel should now show you the CSV file and display the characters correctly.

**Author(s)**

Alessandro Oggioni, PhD (2020) <oggioni.a@irea.cnr.it>

**References**

Scott Chamberlain, Eduard Szocs (2013). "taxize - taxonomic search and retrieval in R." *F1000Research*. <https://f1000research.com/articles/2-191/v2>.

Chamberlain S, Szocs E, Foster Z, Arendsee Z, Boettiger C, Ram K, Bartomeus I, Baumgartner J, O'Donnell J, Oksanen J, Tzovaras BG, Marchand P, Tran V, Salmon M, Li G, Grenié M (2020). *taxize: Taxonomic information from around the web*. R package version 0.9.98, <https://github.com/ropensci/taxize>.

Wickham H, François R, Henry L, Müller K (2022). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.9, <https://CRAN.R-project.org/package=dplyr>.

**Examples**

```
## Not run:
insects <- data.frame(
  taxonID = c(1, 2, 3, 4, 5, 6),
  family = c(
    "Alexiidae", "Anthicidae",
    "Anthribidae", "Anthribidae",
    "Biphyllidae", "Brentidae"
  ),
  scientificName = c(
    "Sphaerosoma seidlitzii", "Endomia tenuicollis tenuicollis",
    "Anthribus fasciatus", "Phaenotherion fasciculatum fasciculatum",
    "Diplocoelus fagi", "Holotrichapion (Apiops) pisi"
  )
)

output <- taxon_id_pesi(
  table = insects,
  taxaColumn = 3
)

# The annotated URIs of columns label are achieved by:
attributes(output)$uri

## End(Not run)
```

---

taxon_id_worms	<i>Enrich and certify a list of species names by comparing with <a href="https://www.marinespecies.org/WoRMS">Rhrefhttps://www.marinespecies.org/WoRMS</a>.</i>
----------------	---

---

**Description**

**[Stable]** This function provide tibble object with all the columns of input table of taxa plus new columns such as valid\_name, valid\_authority, valid\_AphiaID, status, synonyms, LSID, url, matchType, nOfWormsRecords, wormsRecords obtained from Word Register of Marine Species **WoRMS rest API**.

**Usage**

```
taxon_id_worms(input, taxaColumn = 1, verbose = TRUE, refine = FALSE)
```

**Arguments**

input	A tibble. The table that contain the species names list to be checked.
taxaColumn	A numeric. The cardinal number of the column where species list is. Default is 1.

verbose	A logical. With this selection, the function returns a message with number of record(s) that don't match with any Worms names and the number of record(s) that match with more than one Worms name. Default is TRUE.
refine	A logical. With this selection, the function allows to refine the result(s) that match with more Worms records. By an interactive use of the terminal, the user can choose the result. Default is FALSE.

### Value

The output of the function is a tibble with the columns provided and new columns such as: valid\_name, valid\_authority, valid\_AphiaID, status, synonyms, LSID, url, matchType, nOfWormsRecords, wormsRecords obtained by [Worms rest API](#). The function also returns, if verbose is TRUE, the list of records that don't match with Worms name species.

Most of the labels of the columns are the terms of [Darwin Core terms](#). The columns labels are annotated with the link (URI) of the [Darwin Core terms](#) as attributes of the tibble.

### Author(s)

Alessandro Oggioni, PhD (2021) <oggioni.a@irea.cnr.it>

Paolo Tagliolato, PhD (2021) <tagliolato.p@irea.cnr.it>

### References

Chamberlain S (2020). *worms: World Register of Marine Species (WoRMS) Client*. R package version 0.4.2, <https://CRAN.R-project.org/package=worms>.

Wickham H, François R, Henry L, Müller K (2022). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.9, <https://CRAN.R-project.org/package=dplyr>.

### Examples

```

phytoplankton <- tibble::tibble(
  ID = c(1, 2, 3, 4, 5, 6, 7),
  species = c(
    "Asterionella formosa", "Chrysococcus sp.",
    "Cryptomonas rostrata", "Dinobryon divergens",
    "Mallomonas akrokomos", "Melosira varians",
    "Cryptomonas rostrata"
  )
)
table <- taxon_id_worms(
  input = phytoplankton,
  taxaColumn = 2,
  verbose = TRUE,
  refine = TRUE
)
table

# The annotated URIs of columns label are achieved by:
attributes(table)$uri

```

# Index

## \* Internal

plot\_agg\_map, 43  
plot\_timeseries, 44

## \* datasets

package\_settings, 43

## \* package\_customizable\_settings

get\_deims\_base\_url, 8  
package\_settings, 43

elter\_write\_rdata, 3

get\_activity\_info, 4  
get\_dataset\_info, 6  
get\_deims\_API\_version, 8  
get\_deims\_base\_url, 8, 43  
get\_ilter\_envcharacts, 9  
get\_ilter\_generalinfo, 10  
get\_ilter\_observedProperties, 11  
get\_ilter\_research\_topics, 12  
get\_network\_envcharacts, 13  
get\_network\_observedProperties, 15  
get\_network\_related\_resources, 16  
get\_network\_research\_topics, 17  
get\_network\_sites, 18  
get\_sensor\_info, 19  
get\_sensor\_observed\_properties, 21  
get\_site\_info, 22  
get\_site\_MODIS, 24  
get\_site\_ODS, 28  
get\_site\_speciesOccurrences, 30  
get\_sos\_foi, 32  
get\_sos\_obs, 34  
get\_sos\_procedure\_info, 37  
get\_sos\_procedurelist, 36  
get\_zenodo\_data, 39

map\_occ\_gbif2elter, 40  
map\_occ\_inat2elter, 41  
map\_occ\_obis2elter, 42

package\_settings, 8, 43

plot\_agg\_map, 43  
plot\_timeseries, 44  
produce\_network\_points\_map, 45  
produce\_site\_map, 47  
produce\_site\_observedProperties\_pie, 51  
produce\_site\_observedProperties\_waffle, 52  
produce\_site\_qrcode, 54  
produce\_zenodo\_record, 55  
produce\_zenodo\_record\_from\_elter\_reporting, 58

reporting\_compose\_file\_name, 61  
reporting\_produce\_data\_object\_v1.3, 63  
reporting\_save\_archive, 65

save\_occ\_eLTER\_reporting\_Archive, 67  
set\_deims\_base\_url, 68

taxon\_id\_pesi, 68  
taxon\_id\_worms, 70