The Journal of Open Source Software

dataaimsr: An R Client for the Australian Institute of Marine Science Data Platform API which provides easy access to AIMS Data Platform

Diego R. Barneche[1, 2], Greg Coleman[3], Duncan Fermor[3], Eduardo Klein[3], Tobias Robinson[3], Jason Smith[3], Jeffrey L. Sheehan[3], Shannon Dowley[3], Dean Ditton[3], Kevin Gunn[3], Gavin Ericson[3], Murray Logan[3], and Mark Rehbein[3]

1 Australian Institute of Marine Science, Crawley, WA 6009, Australia **2** The Indian Ocean Marine Research Centre, The University of Western Australia, Crawley, WA 6009, Australia **3** Australian Institute of Marine Science, Townsville, Qld 4810, Australia

DOI: 10.21105/joss.03282

Software

- Review
- Repository
- Archive

Editor: Kristen Thyng

Reviewers:

- @kthyng

Submitted: 05 May 2021
Published: 04 June 2021

Summary

`dataaimsr` is an **R package** written to provide open access to decades of field measurements of atmospheric and oceanographic parameters around the coast of Australia, conducted by the Australian Institute of Marine Science (AIMS). The package communicates with the recently-developed AIMS Data Platform API via an API key. Here we describe the available datasets as well as example usage cases.

Statement of Need

The Australian Institute of Marine Science (AIMS) has a long tradition in measuring and monitoring a series of environmental parameters along the tropical coast of Australia. These parameters include long-term record of sea surface temperature, wind characteristics, atmospheric temperature, pressure, chlorophyll-a data, among many others. The AIMS Data Centre team has recently developed the AIMS Data Platform API which is a *REST API* providing JSON-formatted data to users. `dataaimsr` is an **R package** written to allow users to communicate with the AIMS Data Platform API using an API key and a few convenience functions to interrogate and understand the datasets that are available to download. In doing so, it allows the user to fully explore these datasets in R in whichever capacity they want (e.g. data visualisation, statistical analyses, etc). The package itself contains a `plot` method which allows the user to plot summaries of the different types of dataset made available by the API.

Currently, there are two AIMS long-term monitoring datasets available to be downloaded through `dataaimsr`: 1) the Northern Australia Automated Marine Weather And Oceanographic Stations—a list of the weather stations which have been deployed by AIMS and the period of time for which data may be available can be found on the AIMS metadata webpage; 2) AIMS Sea Water Temperature Observing System (AIMS Temperature Logger Program)—for more information on the dataset and its usage, please visit the AIMS metadata webpage.

Technical details and Usage

Before loading the package, a user needs to download and store their personal AIMS Data Platform API Key—we strongly encourage users to maintain their API key as a private,

locally hidden environment variable (`AIMS_DATAPLATFORM_API_KEY`) in the `.Renviron` file for automatic loading at the start of an R session.

`dataaimsr` imports the packages *httr* (Wickham, 2020), *jsonlite* (Ooms, 2014), *parsedate* (Csárdi & Torvalds, 2019), *dplyr* (Wickham et al., 2021), *tidyr* (Wickham, 2021), *rnaturalearth* (South, 2017), *sf* (Pebesma, 2018), *ggplot2* (Wickham, 2016), *ggrepel* (Slowikowski, 2021) and *curl* (Ooms, 2019).

The Weather Station and Sea Water Temperature Loggers datasets are very large (terabytes in size), and as such they are not locally stored. They are instead downloaded via the API and unique DOI identifiers. The datasets are structured by sites, series and parameters. A series is a continuing time-series, i.e. a collection of deployments measuring the same parameter (e.g. Air Temperature, Air Pressure, Chlorophyll) at the same subsite. So, for a given site and parameter, there might exist multiple subsites and therefore series, in which case they are most likely distinguishable by depth.

For the Sea Water Temperature Loggers dataset, series is synonymous with the variable called subsite. For the Weather Station dataset, it is the combination of subsite and parameter.

Discover a dataset

The AIMS Data Platform API points to the full metadata of each dataset. We are currently working on ways to facilitate the visualisation of both datasets and their multiple features directly through the R package. So please consult our on-line vignettes to obtain the most up-to-date instructions on how to navigate the different datasets. Future versions of this package might even provide more of AIMS monitoring datasets.

Data summaries

The first step would be to visualise the dataset. We do this by mapping all available sites. For example, we download the summary information for the Sea Water Temperature Loggers dataset using the main function called `aims_data`. Setting the argument `api_key = NULL` means that `dataaimsr` will automatically search for the user's API key stored in `.Renviron`. The `summary` argument should only be used when the user wants an overview of the available data—this is currently implemented for the Sea Water Temperature Loggers dataset only. One can visualise `summary-by-series` or `summary-by-deployment`. The output of `aims_data` is a `data.frame` of class `aimsdf` with its own plotting method Figure 1:

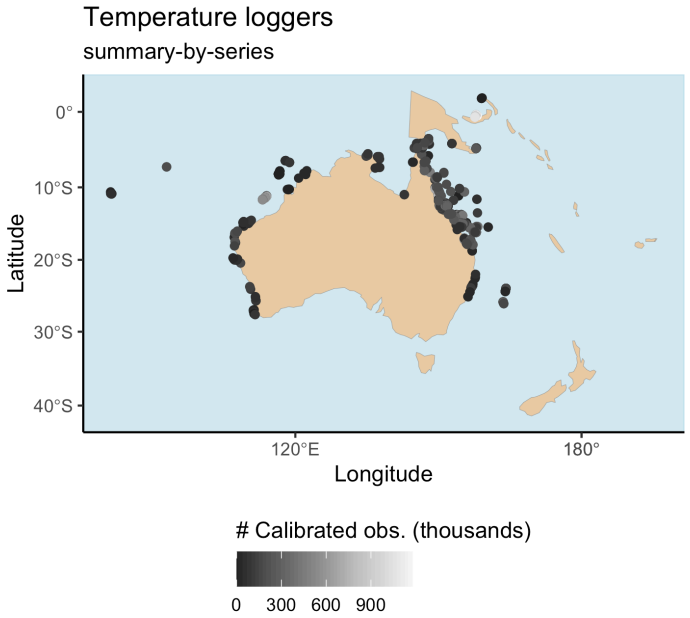

Figure 1: Distribution of all temperature logger series around Australian waters.

For summary data such as `sdata`, plot will always generate a map with the points around Australia and associated regions, coloured by the number of calibrated observations. Observations in a series can be: `uncal_obs`, `cal_obs` and `qc_obs`, which respectively stand for uncalibrated, calibrated, and quality-controlled observations. Calibrated and quality-controlled are generally the same. Instruments are routinely calibrated (mostly once a year) in a temperature-controlled water bath and corrections applied to the data. After calibration, all data records are quality controlled based on the following tests: 1) clip to in-water only data, using deployment's metadata, 2) impossible value check: data outside a fixed temperature range (14 °C – 40 °C) is flagged as bad data, 3) spike test: individual extreme values are flagged as probably bad according to the algorithm presented in Morello et al. (2014) and 4) Excessive gradient test: pairs of data that present a sudden change in the slope are flagged as probably bad (Toma et al., 2016). If any data record fails at least one of the tests, a QC flag equal to 2 is returned, otherwise, the QC flag is set to 1. Please refer to our on-line on-line vignettes to learn details about the entire structure of an `aimsdf` object.

In the case of the Weather Station dataset, the user can call a the `aims_filter_values` function which allows one to query what sites, series and parameters are available for both datasets:

```
head(aims_filter_values("weather", filter_name = "series"))
```

```
##   series_id                                                          ser
## 1    104918        Myrmidon Reef Weather Station Wind Speed (scalar avg b 10 mi
## 2    100686                          Saibai Island Weather Station Hail Durat
## 3       266 Orpheus Island Relay Pole 3 Wind Direction (Vector Average 30 Minut
## 4      2639 Hardy Reef Weather Station Wind Direction (Vector Standard 10 Minut
## 5     10243                          Raine Island Weather Station Air Temperat
## 6       258            Orpheus Island Relay Pole 3 Wind Speed (Scalar avg 10 m
```

The downside is that one cannot know what time window is available for each one of those, nor how they are nested (i.e. series / parameter / site). In a way though the series name

generally gives that information anyway (see code output above). If knowing the available observation window is absolutely crucial, then as mentioned above the user should refer to the on-line metadata.

Download slices of datasets

We recommend slicing the datasets because AIMS monitoring datasets are of very high temporal resolution and if one tries to download an entire series it might take a few hours. To slice the datasets properly, the user needs to apply filters to their query.

Data filters

Filters are the last important information the user needs to know to master the navigation and download of AIMS monitoring datasets. Each dataset can filtered by attributes which can be exposed with the function `aims_expose_attributes`:

```
aims_expose_attributes("weather")
```

```
## $summary
## [1] NA
##
## $filters
##  [1] "site"      "subsite"   "series"    "series_id" "parameter" "size"      "m
```

```
aims_expose_attributes("temp_loggers")
```

```
## $summary
## [1] "summary-by-series"     "summary-by-deployment"
##
## $filters
##  [1] "site"      "subsite"   "series"    "series_id" "parameter" "size"      "m
```

The help file (see `?aims_expose_attributes`) contains the details about what each filter targets. So, having an understanding of the summaries and what filters are available provide the user with a great head start.

Downloading the data is achieved using the same `aims_data` function, however now the `summary` argument is omitted, and instead implement filters. For example, to download all the data collected at the Yongala wreck for a specific time window:

```
wdata_a <- aims_data("weather", api_key = NULL,
                     filters = list(site = "Yongala",
                                    from_date = "2018-01-01",
                                    thru_date = "2018-01-02"))
```

The returned `aimsdf` object in this case has attributes which give us summary crucial information:

- `metadata` a doi link containing the metadata record for the data series
- `citation` the citation information for the particular dataset

- `parameters` an output `data.frame`

These can be directly extracted using the convenience functions `aims_metadata`, `aims_citation` and `aims_parameters`, e.g.:

```
aims_metadata(wdata_a)
```

```
## [1] "Metadata record https://doi.org/10.25845/5c09bf93f315d"
```

This example data contains multiple parameters available for this site at the specified time, and the actual measurements are either raw or quality-controlled. For monitoring data (i.e. when `summary = NA` in a `aims_data` call), we can either visualise the data as a time series broken down by parameter, or a map showing the sites with some summary info. If the parameters are not specified, then `dataaimsr` will plot a maximum of 4 parameters chosen at random for a time series plot. Alternatively the user can specify which parameters are to be plotted Figure 2.

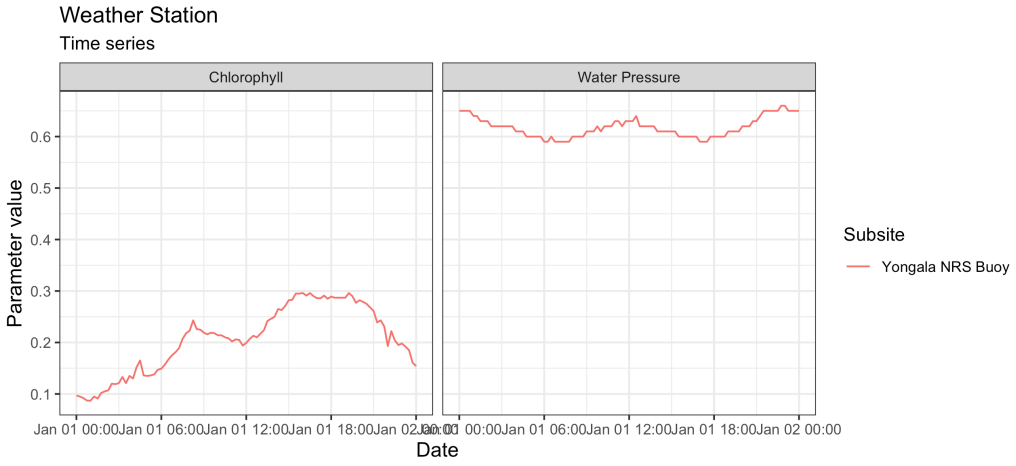

Figure 2: Yongala wreck profiles for water pressure and chlorophyll-a between the first and second of January 2018.

The filters `from_date` and `thru_date` can be further refined by including a time window to download the data:

```
wdata_b <- aims_data("weather", api_key = NULL,
                     filters = list(series_id = 64,
                                    from_date = "1991-10-18T06:00:00",
                                    thru_date = "1991-10-18T12:00:00"))
range(wdata_b$time)
```

```
## [1] "1991-10-18 06:00:00 UTC" "1991-10-18 12:00:00 UTC"
```

Methods

Objects of class `aimsdf` have associated `plot`, `print` and `summary` methods.

Data citation

Whenever using `dataaimsr`, we ask the user to not only cite this paper, but also any data used in an eventual publication. Citation data can be extracted from a dataset using the function `aims_citation`:

```
aims_citation(wdata_b)
```

```
## [1] "Australian Institute of Marine Science (AIMS). 2009, Australian Institute
```

Sister web tool

The Time Series Explorer (https://apps.aims.gov.au/ts-explorer/) is an interactive web-based application that visualises large time series datasets. The application utilises the AIMS Data Platform API to dynamically query data according to user selection and visualise the data as line graphs. Series are able to be compared visually. For large series, data are aggregated to daily averages and displayed as minimum, maximum and mean. When the user 'zooms in' sufficiently, the data will be displayed as non-aggregate values Figure 3. This technique is being used to ensure the application performs well with large time series.

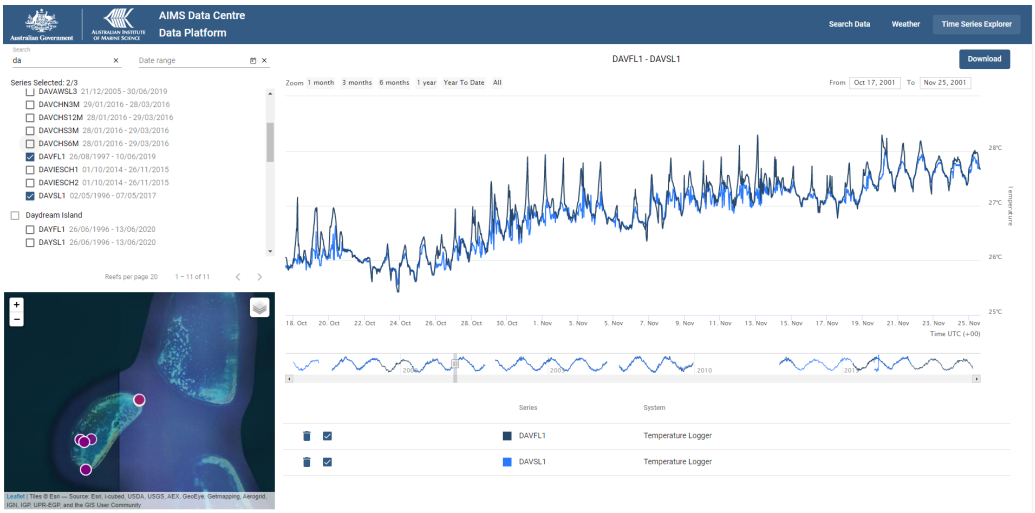

Figure 3: Interactive discovery and visualisation of data series.

The user can then download the displayed data as CSV or obtain a R code snippet that shows how to obtain the data using the dataaimsr package Figure 4. In this way, a user can easily explore and discover datasets and then quickly and easily have this data in their R environment for additional analysis.

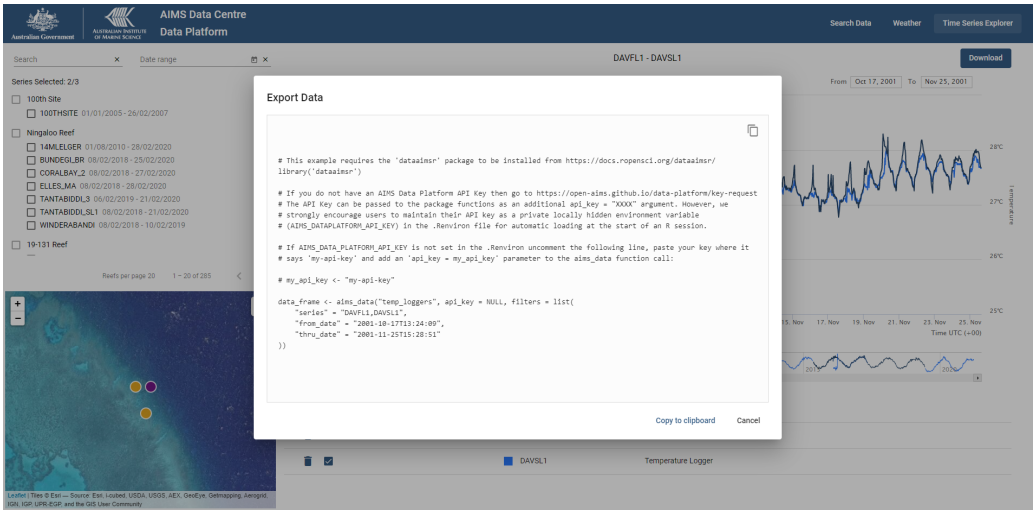

Figure 4: Download/Export displayed data via R snippet.

Future directions

The API is still a work in progress. We are working on ways to better facilitate data visualisation and retrieval, and also we are trying to standardise the outputs from the different datasets as much as possible. In the future, we envision that `dataaimsr` will also provide access to other monitoring datasets collected by AIMS.

References

Csárdi, G., & Torvalds, L. (2019). *Parsedate: Recognize and parse dates in various formats, including all ISO 8601 formats*. https://CRAN.R-project.org/package=parsedate

Morello, E. B., Galibert, G., Smith, D., Ridgway, K. R., Howell, B., Slawinski, D., Timms, G. P., Evans, K., & Lynch, T. P. (2014). Quality Control (QC) procedures for Australia's National Reference Station's sensor data—Comparing semi-autonomous systems to an expert oceanographer. *Methods in Oceanography*, *9*, 17–33. https://doi.org/10.1016/j.mio.2014.09.001

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

Ooms, J. (2019). *Curl: A modern and flexible web client for r*. https://CRAN.R-project.org/package=curl

Pebesma, E. (2018). Simple Features for R: Standardized Support for Spatial Vector Data. *The R Journal*, *10*(1), 439–446. https://doi.org/10.32614/RJ-2018-009

Slowikowski, K. (2021). *Ggrepel: Automatically position non-overlapping text labels with 'ggplot2'*. https://CRAN.R-project.org/package=ggrepel

South, A. (2017). *Rnaturalearth: World map data from natural earth*. https://CRAN.R-project.org/package=rnaturalearth

Toma, D. M., Benadí, A. G., Manuel-Gonzalez, B. J., & del-Río-Fernandez, J. (2016). Systematic quality control for long term ocean observations and applications. *Acta Imeko*, *5*, 64–68. https://doi.org/10.21014/acta_imeko.v5i1.213

Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. Springer-Verlag New York. ISBN: 978-3-319-24277-4

Wickham, H. (2020). *Httr: Tools for working with URLs and HTTP*. https://CRAN.R-project.org/package=httr

Wickham, H. (2021). *Tidyr: Tidy messy data*. https://CRAN.R-project.org/package=tidyr

Wickham, H., François, R., Henry, L., & Müller, K. (2021). *Dplyr: A grammar of data manipulation*. https://CRAN.R-project.org/package=dplyr