

Guide to using the ecoengine R package

The Berkeley Ecoengine (<http://ecoengine.berkeley.edu>) provides an open API to a wealth of museum data contained in the Berkeley natural history museums. This R package provides a programmatic interface to this rich repository of data allowing for the data to be readily analyzed and visualized in a variety of contexts. This vignette provides a brief overview of the package's capabilities.

The API documentation is available at <http://ecoengine.berkeley.edu/developers/>. As with most APIs all requests return a call that displays all the data endpoints accessible to users. Ecoengine has something similar.

```
library(ecoengine)
ee_about()
```

```
## Loading required package: rjson
```

type	endpoint
meta-data	http://ecoengine.berkeley.edu/api/sources/
meta-data	http://ecoengine.berkeley.edu/api/footprints/
data	http://ecoengine.berkeley.edu/api/checklists/
data	http://ecoengine.berkeley.edu/api/sensors/
data	http://ecoengine.berkeley.edu/api/vtmveg/
data	http://ecoengine.berkeley.edu/api/observations/
data	http://ecoengine.berkeley.edu/api/photos/
actions	http://ecoengine.berkeley.edu/api/search/

The ecoengine class

Most functions in the ecoengine package will return a S3 object of class `ecoengine`. The class contains 4 items.

- A total result count (not necessarily the results in this particular object)
- The call (So a reader can replicate the results)
- The type (photos, observation, checklist, or sensor)
- The data. Data are most often coerced into a `data.frame`. To access the data simply use `result_object$data`.

Notes on downloading large data requests

For the sake of speed, results are paginated at 25 results per page. It's possible to request all pages for any query by specifying `page = "all"` in any search. However, this should be used if the request is reasonably sized (1,000 or fewer records). With larger requests, there is a chance that the query might become interrupted

and you could lose all the partially downloaded data. Instead, use the returned observations to split the request.

```
request <- ee_photos()
total_available_observations <- request$results
# This gives you the total number of results available Now divide by 25 to
# get total pages you'll need to request
total_pages <- ceiling(total_available_observations/25)
# Now it's simple to parallelize this request You can parallelize across
# number of cores by passing a vector of pages from 1 through total_pages.
```

Specimen Observations

Photos

```
x <- ee_photos_get(quiet = TRUE)
```

The database currently holds photos. Photos can be searched by state province, county, genus, scientific name, authors along with date bounds. For additional options see `?ee_photos_get`.

Searching photos by author

```
charles_results <- ee_photos_get(author = "Charles Webber")
```

```
## Search returned 4012 photos (downloading page 1 of 161)
```

```
charles_results
```

```
## Number of results: 4012
```

```
## Call: http://ecoengine.berkeley.edu/api/photos/?format=json&page=2&page_size=25&authors=Charles+Webber
```

```
## Output dataset: 25 rows
```

```
# Let's examine a couple of rows of the data
```

```
charles_results$data[1:2, ]
```

```
##          authors          locality          county
```

```
## 1 Charles Webber  Yosemite National Park, Badger Pass Mariposa County
```

```
## 2 Charles Webber Yosemite National Park, Yosemite Falls Mariposa County
```

```
## photog_notes
```

```
## 1      Tan Oak
```

```
## 2      <NA>
```

```
##                                     url
```

```
## 1 http://ecoengine.berkeley.edu/api/photos/CalPhotos%3A8076%2B3101%2B2933%2B0025/
```

```
## 2 http://ecoengine.berkeley.edu/api/photos/CalPhotos%3A8076%2B3101%2B0667%2B0107/
```

```
##   begin_date   end_date          record
```

```
## 1 1954-10-01 1954-10-01 CalPhotos:8076+3101+2933+0025
```

```
## 2 1948-06-01 1948-06-01 CalPhotos:8076+3101+0667+0107
```

```
##                                     remote_resource
```

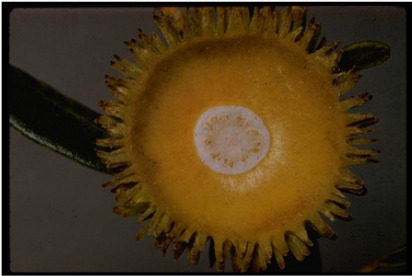
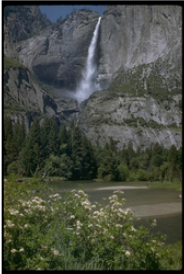
```
## 1 http://calphotos.berkeley.edu/cgi/img_query?seq_num=21272&one=T
## 2 http://calphotos.berkeley.edu/cgi/img_query?seq_num=14468&one=T
##   collection_code observations.scientific_name
## 1      CalAcademy      Lithocarpus densiflorus
## 2      CalAcademy      Rhododendron occidentale
##
##                                     observations.url
## 1 http://ecoengine.berkeley.edu/api/observations/CalPhotos%3A8076%2B3101%2B2933%2B0025%3A1/
## 2 http://ecoengine.berkeley.edu/api/observations/CalPhotos%3A8076%2B3101%2B0667%2B0107%3A1/
##
##                                     media_url
## 1 http://calphotos.berkeley.edu/imgs/512x768/8076_3101/2933/0025.jpeg
## 2 http://calphotos.berkeley.edu/imgs/512x768/8076_3101/0667/0107.jpeg
##
##                                     source geojson.type
## 1 http://ecoengine.berkeley.edu/api/sources/9/          <NA>
## 2 http://ecoengine.berkeley.edu/api/sources/9/          <NA>
##   geojson.coordinates1 geojson.coordinates2
## 1                      <NA>                <NA>
## 2                      <NA>                <NA>
```

Browsing these photos

`view_photos(charles_results)`

This will launch your default browser and render a page with thumbnails of all images returned by the search query.

Ecoengine Photo Viewer

Photo	Authors	Locality / County	Notes	Start Date
	Charles Webber	Yosemite National Park, Badger Pass, Mariposa County	Tan Oak	1954-10-01
	Charles Webber	Yosemite National Park, Yosemite Falls, Mariposa County	NA	1948-06-01

Species checklists There is a wealth of checklists from all the source locations. To get all available checklists:

```
all_lists <- ee_checklists()

## Returning 57 checklists

head(all_lists[, c("footprint", "subject")])

##                                footprint
## 1 http://ecoengine.berkeley.edu/api/footprints/angelo-reserve/
## 2 http://ecoengine.berkeley.edu/api/footprints/angelo-reserve/
## 3 http://ecoengine.berkeley.edu/api/footprints/angelo-reserve/
## 4 http://ecoengine.berkeley.edu/api/footprints/hastings-reserve/
## 5 http://ecoengine.berkeley.edu/api/footprints/angelo-reserve/
## 6 http://ecoengine.berkeley.edu/api/footprints/hastings-reserve/
##      subject
## 1   Mammals
## 2   Mosses
## 3   Beetles
## 4   Spiders
## 5 Amphibians
## 6     Ants
```

Currently there are 57 lists available. We can drill deeper into any list to get all the available data. We can also narrow our checklist search to groups of interest. For example, to get the list of Spiders:

```
spiders <- ee_checklists(subject = "Spiders")

## Returning 2 checklists
```

Now we can drill deep into each list. For this tutorial I'll just retrieve data from the the two lists returned above.

```
library(plyr)
spider_details <- ldply(spiders$url, checklist_details)
names(spider_details)

## [1] "url"                                "observation_type"
## [3] "scientific_name"                  "collection_code"
## [5] "institution_code"                 "country"
## [7] "state_province"                   "county"
## [9] "locality"                         "coordinate_uncertainty_in_meters"
## [11] "begin_date"                       "end_date"
## [13] "kingdom"                          "phylum"
## [15] "class"                            "order"
## [17] "family"                           "genus"
## [19] "specific_epithet"                 "infraspecific_epithet"
## [21] "source"                           "remote_resource"
## [23] "earliest_period_or_lowest_system" "latest_period_or_highest_system"

unique(spider_details$scientific_name)
```

## [1] "holocnemus pluchei"	"oecobius navus"
## [3] "uloborus diversus"	"neriene litigiosa"
## [5] "theridion sp. A"	"tidarren sp."
## [7] "dictyna sp. A"	"dictyna sp. B"
## [9] "mallos sp."	"yorima sp."
## [11] "hahnna sanjuanensis"	"cybaeus sp."
## [13] "zanomys sp."	"anachemmis sp."
## [15] "titiotus sp."	"oxyopes scalaris"
## [17] "zora hespera"	"drassinella sp."
## [19] "phrurotimpus mateonus"	"scotinella sp."
## [21] "castianeira luctifera"	"meriola californica"
## [23] "drassyllus insularis"	"herpyllus propinquus"
## [25] "micaria utahna"	"trachyzelotes lyonneti"
## [27] "ebo evansae"	"habronattus oregonensis"
## [29] "metaphidippus sp."	"platycryptus californicus"
## [31] "calymmaria sp."	"frontinella communis"
## [33] "undetermined sp."	"latrodectus hesperus"
## [35] "theridion sp. B"	"agelenopsis oregonensis"
## [37] "pardosa spp."	"schizocosa mccooki"
## [39] "hololena sp."	"callobius sp."
## [41] "pimus sp."	"aliatypus sp."
## [43] "antrodiaetus sp."	"antrodiaetus riversi"
## [45] "anyphaena californica"	"aculepeira packardi"
## [47] "araneus bispinosus"	"araniella displicata"
## [49] "cyclosa conica"	"cyclosa turbinata"
## [51] "brommella sp."	"cicurina sp."
## [53] "dictyna sp."	"emblyna oregona"
## [55] "orodrassus sp."	"sergiolus sp."
## [57] "erigone sp."	"pityohyphantes sp."
## [59] "tachygyna sp."	"alopecosa kochi"
## [61] "oxyopes salticus"	"philodromus sp."
## [63] "tibellus oblongus"	"pimosa sp."
## [65] "undetermined spp."	"metaphidippus manni"
## [67] "thiodina sp."	"diaea livens"
## [69] "metellina sp."	"cobanus cambridgei"
## [71] "tetragnatha sp."	"tetragnatha versicolor"
## [73] "dipoena sp."	"theridion spp."
## [75] "misumena vatia"	"misumenops sp."
## [77] "tmarus angulatus"	"xysticus sp."
## [79] "hyptiotes gertschi"	"mexigonus morosus"

We've queried data in 80 spider species.

Sensors Some notes on sensors. Where they are located and what kind of data they collect.

```
full_sensor_list <- ee_sensors()
full_sensor_list[, c("station_name", "method_name")]

##           station_name           method_name
## 1 Angelo HQ WS Conversion to 30-minute timesteps
```

```

## 2      Angelo Meadow WS Conversion to 30-minute timesteps
## 3 Angelo HQ SF Eel Gage Conversion to 30-minute timesteps
## 4      Angelo HQ WS Conversion to 30-minute timesteps
## 5      Cahto Peak WS Conversion to 30-minute timesteps
## 6      Angelo Meadow WS Conversion to 30-minute timesteps
## 7      Cahto Peak WS Conversion to 30-minute timesteps
## 8      Angelo HQ WS Conversion to 30-minute timesteps
## 9 Angelo HQ SF Eel Gage Conversion to 30-minute timesteps
## 10 Angelo HQ SF Eel Gage Conversion to 30-minute timesteps
## 11      Angelo Meadow WS Conversion to 30-minute timesteps
## 12      Angelo Meadow WS Conversion to 30-minute timesteps
## 13      Angelo HQ WS Conversion to 30-minute timesteps
## 14      Angelo Meadow WS Conversion to 30-minute timesteps
## 15      Cahto Peak WS Conversion to 30-minute timesteps
## 16      Angelo HQ WS Conversion to 30-minute timesteps
## 17 Angelo HQ SF Eel Gage Conversion to 30-minute timesteps
## 18      Angelo HQ WS Conversion to 30-minute timesteps
## 19      Angelo Meadow WS Conversion to 30-minute timesteps
## 20      Cahto Peak WS Conversion to 30-minute timesteps
## 21      Angelo Meadow WS Conversion to 30-minute timesteps
## 22      Cahto Peak WS Conversion to 30-minute timesteps
## 23      Angelo HQ WS Conversion to 30-minute timesteps
## 24      Angelo Meadow WS Conversion to 30-minute timesteps
## 25      Cahto Peak WS Conversion to 30-minute timesteps
## 26      Angelo Meadow WS Conversion to 30-minute timesteps
## 27      Cahto Peak WS Conversion to 30-minute timesteps
## 28      Angelo HQ WS Conversion to 30-minute timesteps
## 29      Cahto Peak WS Conversion to 30-minute timesteps
## 30      Angelo HQ WS Conversion to 30-minute timesteps
## 31      Cahto Peak WS Conversion to 30-minute timesteps
## 32      Angelo HQ WS Conversion to 30-minute timesteps
## 33      Angelo HQ WS Conversion to 30-minute timesteps
## 34      Angelo Meadow WS Conversion to 30-minute timesteps
## 35      Cahto Peak WS Conversion to 30-minute timesteps
## 36      Angelo Meadow WS Conversion to 30-minute timesteps
## 37      Cahto Peak WS Conversion to 30-minute timesteps
## 38      Angelo HQ WS Conversion to 30-minute timesteps
## 39      Angelo Meadow WS Conversion to 30-minute timesteps
## 40      Angelo Meadow WS Conversion to 30-minute timesteps

```

Searching the engine How to search the engine. Some notes on elastic search.

```

ee_search()
ee_search_obs_get()
ee_search()

```

Aggregated sensor data Ways to obtain aggregated sensor data.

Miscellaneous functions

Footprints

```
footprints <- ee_footprints()
footprints[, -3] # To keep the table from spilling over
```

id	name
12	Angelo Reserve
13	Sagehen Reserve
14	Hastings Reserve
15	Blue Oak Ranch Reserve

Data sources

To obtain a list of data sources for the specimens contained in the museum.

```
source_list <- ee_sources()
unique(source_list$name)
```

unique.source_list.name.
LACM Vertebrate Collection
MVZ Birds
MVZ Herp Collection
MVZ Mammals
Wieslander Vegetation Map
CAS Herpetology
Consortium of California Herbaria
UCMP Vertebrate Collection
Sensor Data Qualifiers
Essig Museum of Entymology

Please send any comments, questions, or ideas for new functionality or improvements to <karthik.ram@berkeley.edu>. The code lives on GitHub [under the rOpenSci account](#). Pull requests and [bug reports](#) are most welcome.

Karthik Ram
January, 2014
Berkeley, CA