gwasrapidd: an R package to query, download and wrangle GWAS Catalog data

Ramiro Magno[1,2] Ana -Teresa Maia[1,2,3,*]

[1] Centre for Biomedical Research (CBMR), Universidade do Algarve, Gambelas Campus, 8005-139 Faro, Portugal
[2] Algarve Biomedical Center (ABC), Universidade do Algarve, Gambelas Campus, 8005-139 Faro, Portugal
[3] Departamento de Ciências Biomédicas e Medicina (DCBM), Universidade do Algarve, Gambelas Campus, 8005-139 Faro, Portugal

* atmaia@ualg.pt

Abstract

Motivation: The NHGRI Catalog of Published Genome-Wide Association Studies (GWAS) Catalog has collected, curated, and made available data from over 3 900 studies. The recently developed GWAS Catalog REST API is the only method allowing programmatic access to this resource.
Results: Here, we describe *gwasrapidd*, an R package that provides a client interface to the GWAS Catalog REST API, representing an important software counterpart to the server-side component. *gwasrapidd* enables users to quickly retrieve, filter and integrate data with comprehensive bioinformatics analysis tools, which is particularly critical for those looking into functional characterisation of risk loci.
Availability: *gwasrapidd* is freely available under an MIT License, and can be accessed from https://github.com/ramiromagno/gwasrapidd.

Keywords

Genome-wide association studies, GWAS, disease susceptibility, Genomics, GWAS Catalog, R, SNP, trait, trait-ontology, Software, human, REST client

Introduction

Over the last 15 years, genome-wide association studies (GWAS) have greatly extended our knowledge of the genetic risk for human complex diseases, and have provided starting points for the understanding of the underlying disease mechanisms, raising possibilities for use in clinical patient/general population stratification [1].

The National Human Genome Research Institute (NHGRI) Catalog of Published GWAS Catalog, created in 2014, is a publicly available, manually curated, database of all published GWA studies [2]. Its latest data release [date 2019-05-03] includes data from 3,989 publications and 138,312 unique SNP-trait associations for human diseases. Currently, these data can be accessed by three methods: (i) via the web graphical user interface, (ii) by downloading database dumps, or, more recently, (iii) via the GWAS

Catalog representational state transfer (REST) application programming interface (API).

Hitherto, the only software that provides parsing of GWAS Catalog data into R is the Bioconductor package *gwascat* [3], which uses the aforementioned database dumps to read in the data. However, the REST API is the only method allowing direct programmatic access, and hence the preferred method for bioinformatics analyses. Yet, so far, no software solution exists to ease the querying and parsing of data returned by it.

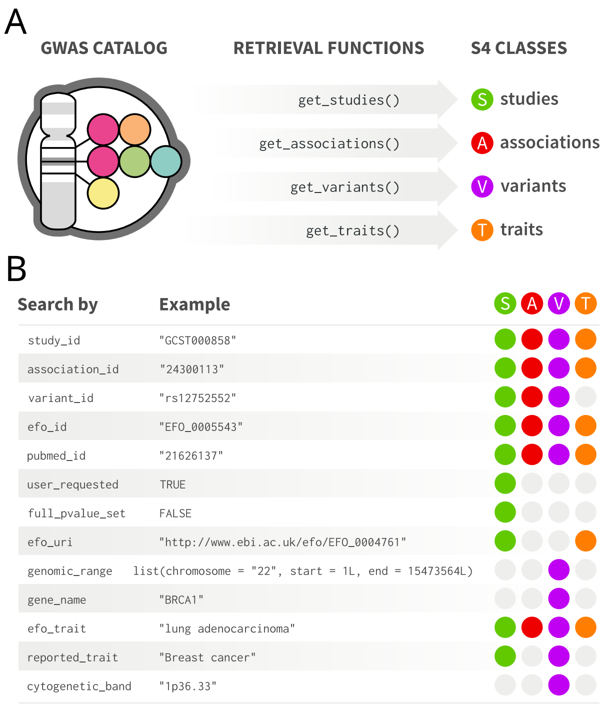
To address this limitation, we have developed an R package [4] that provides programmatic access to the GWAS Catalog REST API: *gwasrapidd*. This package provides a simple interface for querying Catalog data, abstracting away the informatic details of the REST API. In addition, retrieved data is mapped to in-memory relational databases of tidy data tables, allowing prompt integration with tidyverse packages for subsequent transformation, visualisation and modelling of data [5, 6].

Results

Retrieving data from the GWAS Catalog REST API

The GWAS Catalog REST API is an EBI service hosted at `https://www.ebi.ac.uk/gwas/rest/api/`. The REST API uses hypermedia with resource responses following the JSON Hypertext Application Language (HAL) format [7]. Response data is therefore provided as hierarchical data in JSON format. Although flexible, hierarchical data is not straightforwardly convertible to tabular format — the preferred format for data analysis in R [6]. Moreover, a resource response can be paginated, i.e., split into multiple responses, requiring navigation over all the individual responses, requiring posterior aggregation. Finally, data can also be embedded, i.e., have other resources contained within them, adding extra complexity to the returned JSON format (Additional file 1: Table S1, and [8]).

To ease the conversion from the hierarchical to the relational tabular format, and to abstract away the informatic details associated with the HAL format, we developed a set of retrieval functions (Fig. 1A). Since the REST API data is organised around four core data entities —*studies*, *associations*, *variants* and *traits* [8]— we implemented four corresponding retrieval functions that encapsulate the technical aspects of resource querying and format conversion: `get_studies`, `get_associations`, `get_variants` and `get_traits` (Fig. 1A). These functions simplify the querying of GWAS entities, by providing a complete and consistent interface to the Catalog. For example, to query for *studies*, the user needs only to know the function `get_studies`, whereas the REST API itself exposes a set of disparate resource URL endpoints for *studies* following the available search criteria (Additional file 1: Table S1, and [8]). Moreover, the user can choose from any number of available search criteria exposed by the REST directly as arguments to the retrieval functions (Fig. 1B). All arguments are vectorised, meaning that multiple queries are promptly available from a single function call. Results obtained from multiple queries can be combined in an `OR` or `AND` fashion with the `set_operation` parameter. If `set_operation` is set to `OR` (default behaviour), results are collated while removing duplicates, if any. If `set_operation` is set to `AND`, only entities that concomitantly match all criteria are returned. If finer control is needed on combining results from different queries, the user can make separate calls with the retrieval functions and combine the results from the different objects with any combination of these functions: `bind()`, `union()`, `intersect()`, `setdiff()` and `setequal()`. These are S4 methods that work with the S4 classes created in *gwasrapidd* (Fig. 2).



Search by	Example	S	A	V	T
study_id	"GCST000858"	●	●	●	●
association_id	"24300113"	●	●	●	●
variant_id	"rs12752552"	●	●	●	○
efo_id	"EFO_0005543"	●	●	●	●
pubmed_id	"21626137"	●	●	●	●
user_requested	TRUE	●	○	○	○
full_pvalue_set	FALSE	●	○	○	○
efo_uri	"http://www.ebi.ac.uk/efo/EFO_0004761"	●	○	○	●
genomic_range	list(chromosome = "22", start = 1L, end = 15473564L)	○	○	●	○
gene_name	"BRCA1"	○	○	●	○
efo_trait	"lung adenocarcinoma"	●	●	●	●
reported_trait	"Breast cancer"	●	○	●	○
cytogenetic_band	"1p36.33"	○	○	●	○

Figure 1. *gwasrapidd* retrieval functions. **(A)** Functions for retrieving data from the GWAS Catalog: `get_studies()`, `get_associations()`, `get_variants()` and `get_traits()`. **(B)** *gwasrapidd* search criteria (function parameters) to be used with retrieval functions. Coloured circles indicate which entities can be retrieved by which criteria.

Representation of GWAS Catalog entities

The GWAS Catalog REST API provides the data organised according to four key concepts: *studies*, *associations*, *variants* and *traits*; reflected on the possible types of JSON responses. In *gwasrapidd*, each of these four types of responses to S4 objects are mapped to four classes of the same name: `studies`, `associations`, `variants` and `traits`, respectively (Fig. 2). All S4 classes share the same design principles that makes them relational databases: (i) each slot corresponds to a table (dataframe in R), (ii) the first slot corresponds to the main table that lists observations of the respective GWAS Catalog entity, e.g., *studies*, and, (iii) all tables have a primary key, the identifier of the respective GWAS Catalog entity: `study_id`, `association_id`, `variant_id` or `efo_id` (Fig. 2). For easy consultation of the variables in the tables, we provide a cheatsheet (Additional file 2: gwasrapidd cheatsheet); for the detailed description the user can issue the following commands to open the help page about each class: `class?studies`, `class?associations`, `class?variants` or `class?traits`.

Case study

To illustrate how to use *gwasrapidd*, we take as a motivating example the work by Light *et al.* [9]. In this work the authors aimed at characterising chromatin status at allelic resolution, as a strategy for elucidanting the cis-regulatory mechanisms behind complex disease risk. To perform this characterisation, the authors started by selecting variants previously reported in the GWAS Catalog for autoimmune disease. We enact now what could have been these first steps of their approach using *gwasrapidd*. We start by loading *gwasrapidd*:

```r
library(gwasrapidd)
```

Then following by querying the GWAS Catalog for *studies* by searching by *autoimmune disease* (an EFO trait):

```r
my_studies <-
  get_studies(efo_trait = 'autoimmune disease')
```

One can now check how many GWAS studies were retrieved using function `n()`. The same function could be used for the other entities: `associations`, `variants` or `traits`.

```r
n(my_studies)
#> [1] 1
my_studies@studies$study_id
#> [1] "GCST003097"
```

Seemingly, only one study matched exactly `'autoimmune disease'`: the study with the identifier `GCST003097`. We can inspect the original publication(s) that underlie this GWA study entry in the Catalog. For example, to access the associated publication title one can access the `title` variable from the `publications` table:

```r
my_studies@publications$title
#> [1] "Meta-analysis of shared genetic architecture across ten pediatric
↪   autoimmune diseases."
```

To quickly browse to the PubMed entry for this publication, the user may use the helper function `open_in_pubmed()`:

```r
# This launches your web browser at https://www.ncbi.nlm.nih.gov/pubmed/26301688
open_in_pubmed(my_studies@publications$pubmed_id)
```

Now if we want to know the variants previously associated with autoimmune disease, as used by Light et al. (2014), we need to retrieve statistical association information on these variants, and then apply a filter using the same level of significance $P < 1\times10^{-6}$ [9]. A quick inspection at the *gwasrapidd* cheatsheet (Additional file 2: gwasrapidd cheatsheet) hints that statitiscal information, such as p-values and odds ratios can be found in the `associations` table of the `associations` S4 class object. So, we can get the associations by the previously found *study* identifier (`GCST003097`):

```r
# Alternative query that would work too:
# get_associations(
#   efo_trait = 'autoimmune disease'
#   )
my_associations <- get_associations(study_id = "GCST003097")
```

We find 46 *associations*:

```r
n(my_associations)
#> [1] 46
```

However, it might be that not all variants meet the level of significance, as required by Light et al. (2014). We use now functions from the *dplyr* package [10] to filter the table `associations` from the S4 object `associations` based on the p-value (`pvalue` column).

```r
# Get association ids
# for which pvalue is less than 1e-6.
dplyr::filter(
  my_associations@associations,
  pvalue < 1e-6) %>%
  dplyr::pull(association_id) ->
  association_ids
```

Having the *association* identifiers (`association_ids`) that meet the requirement $P < 1 \times 10^{-6}$, we can easily create a new S4 object (`my_associations2`) containing only those *associations* using the subset operator `` `[` ``:

```r
# Extract associations by association id
my_associations2 <- my_associations[association_ids]
```

The subset operator `` `[` `` can also be used with `integer` values to subset by position. Note that both ways of subsetting will filter all tables within the S4 object for the matched identifiers. Now we can check how many associations are still present in `my_associations2`:

```r
n(my_associations2)
#> [1] 44
```

Of the 46 associations found in GWAS Catalog, 44 meet the p-value threshold of 1×10^{-6}. Finally, to see variants, we just need to access the table `risk_alleles` from the `my_associations2` object. Here are the variant identifers, risk alleles and risk frequency:

```r
my_associations2@risk_alleles[c('variant_id', 'risk_allele', 'risk_frequency')]
#> # A tibble: 44 x 3
#>    variant_id  risk_allele risk_frequency
#>    <chr>       <chr>                <dbl>
#>  1 rs2066363   C                     0.34
#>  2 rs114846446 A                     0.01
#>  3 rs7672495   C                     0.18
...
```

Discussion

We have developed an R client to the GWAS Catalog REST API, thus allowing programmatic access to the database. The main features of *gwasrapidd* are: (i) abstracting away the REST API informatic details by providing a simple and consistent interface, and (ii) a tidy data representation of the GWAS entities, i.e., of *studies*, *associations*, *variants* and *traits* in the form of in-memory relational databases. Below, we highlight some of the improvements and limitations of our package.

Improvements

Compared to the exposed REST API, we have augmented the search possibilities in *gwasrapidd* in two ways: (i) by allowing the user to search for *variants* by cytogenetic region (as is possible with the web GUI), and (ii) by allowing searching *variants* by EFO identifier (`efo_id`). Searching by cytogenetic region was implemented by

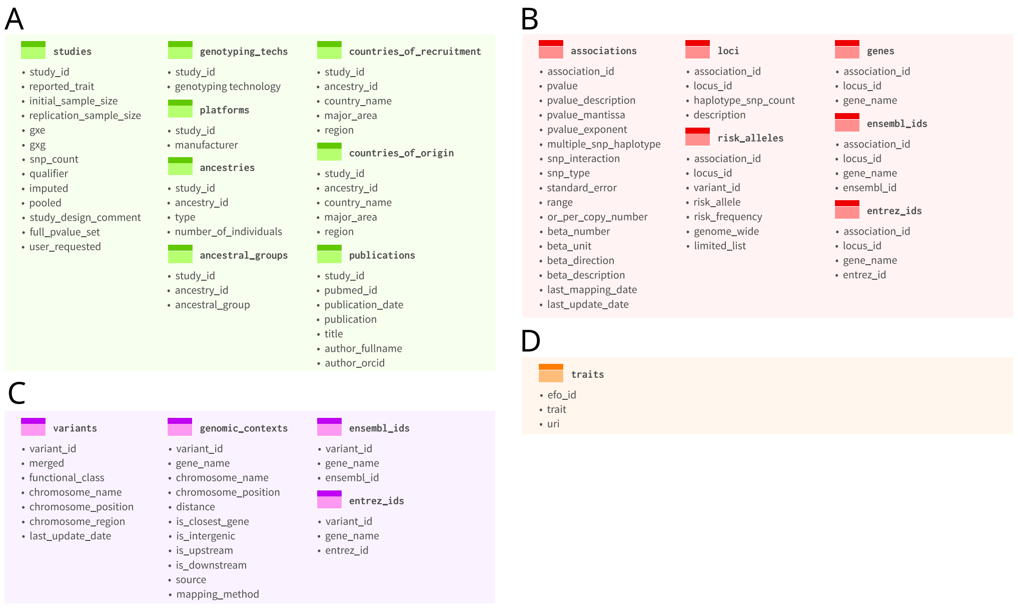


Figure 2. S4 representation of *studies*, *associations*, *variants* and *traits*. **(A)** S4 `studies` object, comprising 8 slots, i.e., 8 tables: `studies`, `genotyping_techs`, `platforms`, `ancestries`, `ancestral_groups`, `countries_of_recruitment`, `countries_of_origin` and `publications`. All tables share the primary key `study_id`. **(B)** S4 `associations` object, comprising 6 slots, i.e., 6 tables: `associations`, `loci`, `risk_alleles`, `genes`, `ensembl_ids` and `entrez_ids`. All tables share the primary key `association_id`. **(C)** S4 `variants` object, comprising 4 slots, i.e., 4 tables: `variants`, `genomic_contexts`, `ensembl_ids` and `entrez_ids`. All tables share the primary key `variant_id`. **(D)** S4 `traits` object, with one slot, i.e., one single tables: `traits`. The `traits` has the primary key `efo_id`.

embedding a dataset of genomic ranges of the human cytogenetic bands in *gwasrapidd*. In this way, queries made with cytogenetic band names can be translated to genomic ranges, which are then used with the exposed interface of the REST API to search by genomic range (`genomic_range`). Searching *variants* by EFO identifier (`efo_id`) is not directly available in the current version of the REST API. Therefore, to create this possibility in *gwasrapidd*, we map EFO identifiers (`efo_id`) to EFO traits with `get_traits()` using the former as a search parameter. The table `traits` of the S4 object `traits` contains the `trait` column, providing a one-to-one mapping between each EFO identifier and its trait description (`trait`). Then, we use internally `get_variants()` to search by EFO trait description (`efo_trait`).

Finally, *gwasrapidd* also provides a set of helper functions to easily browse linked web resources, such as PubMed [11] , dbSNP [12],and GTEx project [13].

Limitations

The most popular method of access to the Catalog is still the web graphical user interface (GUI), and is therefore still the one that gets more attention and dedication from the GWAS Catalog team. Thus, compared to the web GUI, the REST API is still lagging behind in functionality. Here we discuss some of the limitations of the REST API compared to the web GUI.

When searching by publication related information, with the REST API, we can only use the PubMed identifier as a query. Thus, to search using other publication related criteria, such as words in the title or date of publication, one needs to download all studies first with `get_studies` and then perform filtering based on variables from the `publications` table of the `studies` object; thus, albeit possible, this introduces unnecessary delays impacting the user experience. Using the web GUI one can promptly use the free text search bar to query publication titles, or dates.

Another important type of query that benefits from the free text search bar is searching by trait. Currently, with the REST API one can search by EFO identifier (`efo_id`) or by EFO trait description (`efo_trait`) with all retrieval functions (Fig. 1B). The author's reported trait (`reported_trait`) can only be used to retrieve *studies* or *variants*. In both cases, the search terms need to be exact matches. Conversely, with the web GUI one can use free text to get first a list of possible traits and then find other entities such as *studies* and *associations* using the EFO trait of interest. Again, this is still possible with *gwasrapidd*, but indirect. One can retrieve all traits first with `get_traits` and then filter by the `trait` using regular expressions to get all traits related to a specific term or collection of terms, and then use any of the four retrieval functions with the `efo_trait` parameter. The web GUI also allows to include child trait terms when searching by trait. This is not currently possible with the REST API.

Another limitation with the REST API is that its database release date is older than the web GUI version. As of this writing, the REST API provided data was last updated in 2019-01-12, whereas the web GUI data is dated 2019-04-21. The REST API database release version can be retrieved with the function `get_metadata()`.

Conclusions

We have developed the first R client to the GWAS Catalog REST API, thus greatly facilitating programmatic access to the database. This improves data mining from within R, accelerating the integration of GWAS data into further genomic and biomedical/clinical studies. *gwasrapidd* is freely available under the MIT Licence and can be accessed from `https://github.com/ramiromagno/gwasrapidd`.

Availability and requirements

- **Project name**: gwasrapidd.
- **Project home page**: `https://rmagno.eu/gwasrapidd`.
- **Operating system(s)**: Platform independent.
- **Programming language**: R.
- **Other requirements**: None.
- **License**: MIT.
- **Any restrictions to use by non-academics**: None.

Competing interests

The authors declare that they have no competing interests.

Funding

This work was supported by national Portuguese funding through FCT—Fundação para a Ciência e a Tecnologia and CRESC ALGARVE 2020: UID/BIM/04773/2013 "CBMR", POCI-01-0145-FEDER-022184 "GenomePT" and ALG-01-0145-FEDER-31477 "DEvoCancer".

Author's contributions

RM devised and wrote the package, and wrote the manuscript. ATM supervised the project and wrote the manuscript. Both authors read and approved the final manuscript.

Acknowledgements

We would like to thank the GWAS Catalog team, particularly Daniel Suveges, for all the help and support with the REST API throughout the entire development of *gwasrapidd*. We thank also the remaining of the Cancer Functional Genomics lab members for feedback on the user experience of *gwasrapidd*.

References

1. Visscher PM, Wray NR, Zhang Q, Sklar P, McCarthy MI, Brown MA, et al. 10 years of GWAS discovery: biology, function, and translation. The American Journal of Human Genetics. 2017;101(1):5–22.
2. Welter D, MacArthur J, Morales J, Burdett T, Hall P, Junkins H, et al. The NHGRI GWAS Catalog, a curated resource of SNP-trait associations. Nucleic Acids Research. 2013;42(D1):D1001–D1006.
3. Carey V. gwascat: representing and modeling data in the EMBL-EBI GWAS Catalog; 2018. R package version 2.14.0.
4. R Core Team. R: A Language and Environment for Statistical Computing. Vienna, Austria; 2017. Available from: https://www.R-project.org/.
5. Wickham H, et al. Tidy data. Journal of Statistical Software. 2014;59(10):1–23.
6. Wickham H, Grolemund G. R for Data Science: Import, Tidy, Transform, Visualize, and Model Data. 1st ed. O'Reilly Media; 2017. Available from: http://r4ds.had.co.nz/.
7. Mike Kelly. JSON Hypertext Application Language; 2016. [Online; accessed 04-May-2019]. https://tools.ietf.org/html/draft-kelly-json-hal-08/.
8. NHGRI-EBI GWAS Catalog Team. GWAS Catalog API Guide; 2019. [Online; accessed 04-May-2019]. https://www.ebi.ac.uk/gwas/rest/docs/api/.
9. Light N, Adoue V, Ge B, Chen SH, Kwan T, Pastinen T. Interrogation of allelic chromatin states in human cells by high-density ChIP-genotyping. Epigenetics. 2014;9(9):1238–1251.
10. Wickham H, François R, Henry L, Müller K. dplyr: A Grammar of Data Manipulation; 2019. R package version 0.8.0.1. Available from: https://CRAN.R-project.org/package=dplyr.

11. Lindberg D. Internet access to the National Library of Medicine. Effective clinical practice: ECP. 2000;3(5):256.

12. Sherry ST, Ward MH, Kholodov M, Baker J, Phan L, Smigielski EM, et al. dbSNP: the NCBI database of genetic variation. Nucleic Acids Research. 2001 Jan;29(1):308–311.

13. GTEx Consortium. The Genotype-Tissue Expression (GTEx) project. Nature Genetics. 2013 Jun;45(6):580–585.

Additional Files

Additional file 1 — Table S1. GWAS Catalog REST API URL endpoints.

- **File name**: `additional_file_1.pdf`.
- **File format**: Portable Document Format (PDF).
- **Title**: Table S1. GWAS Catalog REST API URL endpoints.
- **Description**: Additional file 1 contains Supplementary Table 1. Table S1 describes the GWAS Catalog REST API endpoints, along with its parameters (search criteria), and attainable GWAS entities: *studies*, *associations*, *variants* and *traits*.

Additional file 2 — gwasrapidd cheatsheet.

- **File name**: `additional_file_2.pdf`.
- **File format**: Portable Document Format (PDF).
- **Title**: gwasrapidd cheatsheet.
- **Description**: Additional file 2 contains an infographics: gwasrapidd cheatsheet.