The Journal of Open Source Software

Vigicaen: A `vigibase`® Pharmacovigilance Database Toolbox.

Charles Dolladille [1] and **Basile Chrétien** [2]

1 University of Caen Normandy, Pharmacology Department, Centre Hospitalier Universitaire de Caen, Caen, France **2** University of Nagoya, Department of biostatistics, Nagoya University Hospital, Nagoya, Japan

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 25 August 2025
Published: unpublished

Summary

Advanced methodologies are essential when conducting disproportionality analyses using pharmacovigilance data, as traditional approaches are susceptible to various biases such as reporting bias and confounding. The aim of vigicaen is to provide a toolbox for the VigiBase® Extract Case Level database, resolving technical challenges related to the database large size, and providing easier and reproducible access to advanced features. The package is built on top of the parquet file format. Functions related to drug and adverse event identification, descriptive features such as time to onset, dechallenge and rechallenge outcomes are provided. Command line side-effect outputs aim at fast resolving of common issues related to drug and adverse event identification. The package is intended for pharmacovigilance practitioners, clinicians and researchers with or without advanced biostatistical skills. A graphical output can be produced for routine use, to support daily assessment of drug liability.

Statement of need

Disproportionality analysis represents an essential component in the domain of drug safety signal detection. Advanced methodologies are required to address common biases within pharmacovigilance databases. These analyses necessitate expertise in biostatistical software, such as R, which may present substantial challenges in terms of acquiring and maintaining the requisite skills — in addition to a solid understanding of pharmacovigilance principles and reporting systems.

For decades, the World Health Organization (WHO) has been collecting adverse drug reaction reports, called Individual Case Safety Reports (ICSRs), from its member countries, populating more than 40 millions reports to date. This pharmacovigilance database is called VigiBase® and is managed by the Uppsala Monitoring Centre in Sweden.(Centre, n.d.) These ICSRs describe the course of patients who experienced an adverse event (a medical condition) after taking a drug. The burning question is whether this adverse event was actually related to the drug intake, e.g. if it is an adverse drug *reaction* (ADR). The pharmacovigilance database aims at uncovering the very first potential signals of association between drugs and ADRs.(Montastruc et al., 2011)

It relies on disproportionality analysis, a statistical method that produces estimators of how unlikely the number of observed ICSRs reporting on a specific drug and adverse event is to be attributable to chance alone. Together with an incertitude margin, these estimators are used to raise safety signals on drugs.(Montastruc et al., 2011)

The Uppsala Monitoring Centre grants access to VigiBase® to researchers, either academics or industrials, under a licence contract. The most extensive available version is called Extract Case Level: It contains all the ICSRs, with information such as the patient demographics, the drug intake, the adverse events, the outcome, the dechallenge and rechallenge outcome, and the time to onset. However, this version is provided as large text files, and requires a lot of processing before being usable for analysis. Those text files might be particularly challenging to use in R, as they would often exceed the size of the available Random Access Memory, thus requiring advanced knowledge of R computing techniques. Clinicians and pharmacovigilance practitioners typically lack these skills, and therefore struggle to use the VigiBase® data for their research. As a result, they would often rely on partial data, with limited statistical modelling options. Or, they could develop home-made biostatistic scripts that would typically be used once, often left undocumented, and highly heterogeneous across research teams.

The vigicaen package aims at providing a toolbox for the VigiBase® Extract Case Level database, tackling a few technical challenges to run on low-specification computers, and provide easy and reproducible access to advanced features.(Dolladille & Chrétien, 2025) This article will explain the technical choices and data management logic underlying the package, and provide some examples of its main features. Additional examples and use cases are treated in the package vignettes, which can be found on the package website at https://pharmacologie-caen.github.io/vigicaen/. Of important note, the package is not supported nor reflects the opinion of the WHO. The Uppsala Monitoring Centre, in charge of maintaining VigiBase®, was informed of the package development and kindly allowed its publication, acknowledging the potential benefit to promote the use of VigiBase®.

Research impact and significance

Our team and collaborators have already published several pharmacovigilance studies using vigicaen. Minoc et al. (2025). Vigicaen streamlines the data management process of pharmacovigilance studies, allowing for easier collaboration across centers around the world. The potential gain has already convinced several academic centres. The French Network of Regional Pharmacovigilance Centres is on its way to implement vigicaen as part of the routine practice across the 31 Pharmacovigilance Centres in France. The University of Nagoya has functional routines relying on vigicaen for disproportionality analyses. Vigicaen is not getting in any concurrence with existing open source tools, but rather addresses an unmet need.

Design thinking

Key concepts were fundamental to building vigicaen: First, it should be open source, build on top of state-of-the-art practices to deal with large datasets (e.g. arrow), especially on low specification computers, using widespread and consistent syntax R users would be familiar with (e.g. tidyverse). Although other syntaxes like `data.table` were once at the core of the package, they are now generally left over, as they were thought less fit to the project when considering the balance between performance and user facing interace. Second, it should address the most technically challenging issues for beginners in R or biostatistical softwares in general. Third, it should keep as much rigor and consistency as possible in the function naming, expected input formats, and outputs. Fourth, it should provide help, e.g. messages to users, to allow external checking of what is produced by the package. Fifth, it is not purposed to implement model functions (like `glm`) per se, but rather to prepare the dataset so as to let the user run any model of his/her choice. Simple computations are nevertheless in the scope (like bivariate disproportionality analysis, or basic interaction analysis).

Open-source software practice

The package was developed according to best practices as promoted by R Packages, 2nd edition. *R Packages (2e)* (n.d.) It is accompanied by a comprehensive set of unitary tests (covering 100% of the code), in-depth documentation for each function and object, and several tutorial vignettes for both new-comers and advanced users. The source code is available on GitHub.com, so as to provide a unique platform to submit issues and propose pull requests. It is made available under the open source CeCILL 2.1 license.

Development history

The first iteration of the package was built in 2020, as a local software designed for internal use at Caen University Hospital. Later, it was called pharmacocaen and posted as a private repository on GitHub on 2022, due to intellectual property concerns with Uppsala Monitoring Centre. After resolution of property concerns, the package became available as a public repository on GitHub under the name vigicaen in 2024, and was accepted on CRAN in 2025. In the first versions, the package was mainly focused on performing vectorized data management so as to identify a large number of drugs and reactions in a compiled way. As there was a wide variety of settings under which drugs and reactions could be identified, bug fixing and edge cases were the main concerns during several years. Then, additional functionalities like building datasets from text files and descriptives were added. Contacts were made with members from the Uppsala Monitoring Centre, regarding their own work on other topics. These exchanges helped defining the exact perimeter of vigicaen, as well as its potential articulation with other open source softwares in the future. Also, vigicaen was discussed with end-users from pharmacovigilance centres in France, which led to the development of specific functions like `vigi_routine`.

Processing `vigibase`® source files.

Clinicians and pharmacovigilance researchers are used to work with low-specification computers. The typical available Random Access Memory rarely exceeds 16GB, which is one of the key resources to deal with large data files in R.(*22 Arrow – r for Data Science (2e)*, n.d.) VigiBase® Extract Case Level files currently exceed 30GB once unpacked, which is way too large to be loaded in-memory for mainstream readers like `read.table()`.

Vigicaen relies on `parquet` files a recent format based on open standards.(*Parquet*, n.d.) Arrow is a cross-language development platform that allows for manipulation of large datasets.(*Apache Arrow*, n.d.) It is implemented in R via the arrow package.(Richardson et al., 2025) Datasets remain out of memory, allowing for processing of large files on low-specification computers. Various tests of vigicaen on 16GB RAM computers succeeded in processing the source files. This, in combination with an as close as possible alignment with the tidyverse style guide, is also aimed at providing a modern and more rigorous approach as compared to base R.(Wickham & RStudio, 2023)

Sourcing VigiBase® Extract Case Level files is done with the `tb_*` family functions.

First, we define paths to the source folders.

```
library(vigicaen)

path_base <- paste0(tempdir(), "/main/")
path_sub  <- paste0(tempdir(), "/sub/")

dir.create(path_base)
dir.create(path_sub)
```

Example files can be put in these folders.

```
create_ex_main_txt(path_base)
create_ex_sub_txt(path_sub)
```

Then, we run the related `tb_*` function, `tb_vigibase()`.

```
tb_vigibase(path_base, path_sub)
```

```
##
## -- tb_vigibase() ---------------------------------------------------------------
## i Checking for existing tables.
## i Creating vigibase tables.
## This process must only be done once per database version.
## It can take up to 30minutes.
## =========>---------------------   31% | 1.1s | Read SRCE.txt
## ==========>--------------------   34% | 1.2s | Write srce.parquet
##
```

With an average computer, the real running time is around 20-30minutes on current database version.

If the dictionaries for drugs and adverse events are also required, `tb_who()` and `tb_meddra()` can be used.

Identifying drugs and adverse events

Exposure to drugs and occurrence of adverse events are located in the `drug` and `adr` tables, respectively. They connect together through the `demo` table, in a many-to-one relationship, via the `UMCReportId` key variable. Drugs and adverse events themselves are identified by codes (or IDs) from the WHO Drug Dictionary and the Medical Dictionary for Regulatory Activities (MedDRA), respectively. Disproportionality analysis requires a dataset with one row per ICSR, with the corresponding drugs and adverse events.

The following logic is implemented in vigicaen:

1 Use drug and adverse event names to collect their IDs.

2 Match the IDs in `drug` and `adr` tables to identify the cases.

3 Report this information in `demo` (or any other VigiBase® table).

This is done with the `get_*` functions (step 1), and the `add_*` functions (steps 2 and 3). The overall process requires the sequential use of both. Below is an example to identify the drugs. The same principle is applied to adverse events.

```r
# load vigibase tables and drug dictionary
demo <- dt_parquet(path_base, "demo")
drug <- dt_parquet(path_base, "drug")

# for the demonstration, we will use built-in example files
demo <- demo_
drug <- drug_
mp   <- mp_

# Select drug names
d_sel <-
  list(ipilimumab = "ipilimumab")

# Get the drug IDs
d_drecno <-
  get_drecno(
    d_sel,
    mp = mp
  )
```

```
##
## -- get_drecno() ---------------------------------------------------------------
##
## -- `d_sel`: Matching drugs --
##
## -- v Matched drugs
##
## > `ipilimumab`: "ipilimumab" and "ipilimumab;nivolumab"
##
##
## i Set `verbose` to FALSE to suppress this section.
##
##
##
## --------------------------------------------------------------------------------
```

```
# report into demo
demo <-
  demo |>
  add_drug(
    d_drecno,
    drug_data = drug
  )
```

```
## i `.data` detected as `demo` table.
```

Displaying information at the command line

As seen in the output above, the get_* functions do 2 things: They return drug or adverse event IDs (stored in d_drecno in the example), and they display command line information about the matching process. This is especially useful since drugs and adverse events name may vary in their spelling and case, while the underlying dictionary only accepts exact matches. Matched and un-matched names are displayed, along with some hints for the unmatching reasons.

```
meddra <- meddra_

a_sel <-
  list(colitis_term = c("Colitis", "Autoimmune colitis"),
       pneumonitis_term = "pneumonitis")

a_llt <- get_llt_soc(a_sel, term_level = "pt", meddra = meddra)
```

```
##
## -- get_llt_soc() ---------------------------------------------------------------
##
## -- v Matched reactions at `pt` level (number of codes) --
##
```

```
## > `colitis_term`: "Autoimmune colitis (1)" and "Colitis (25)"
## > `pneumonitis_term`: x No match
##
##
## i Set `verbose` to FALSE to suppress this section.
##
##
##
## -- x Unmatched reactions --
##
##
##
## -- ! Some reactions did not start with a Capital letter
##
##
##
## * In `pneumonitis_term`: x "pneumonitis"
```

The named list for inputting drug and adverse event names

The get_* and add_* functions are built on top of named list as first argument. This structure may seem a bit busy, especially for new comers, but it allows for genuine flexibility when analyses plan increments. As an example, one may create `list(drug_group_1 = c("ipilimumab", "nivolumab"))` to automatically gather all ICSRs reporting one of these two drugs, through `get_drecno()` and `add_drug()`.

Descriptive features

Descriptive features often take an important place in pharmacovigilance studies. They may be as important as producing statistical estimands, to assess the liability of a given drug. Among them, the time to onset is rather challenging to compute. The main reasons are the incertitude around the exact reported time to onset, and the potential multiple reports for a given drug-adverse event pair in a single ICSR. The first is tackled by the Uppsala Monitoring Centre, which recommends in internal documentation to analyze ICSR where the incertitude interval is no more than a day. The second is addressed in `extract_tto()` or `desc_tto()`, which only extracts the longest time to onset reported for a given drug-adverse event pair in a given ICSR. This variable is called `tto_max`. Admittedly, this is a simplification that might not cover all potential use cases, for example if the question is the time since last infusion of a drug.

A similar simplifying approach is applied to drug dechallenge (`desc_dch()`) and rechallenge (`desc_rch()`) outcomes, as well as adverse event outcome (`desc_outcome()`).

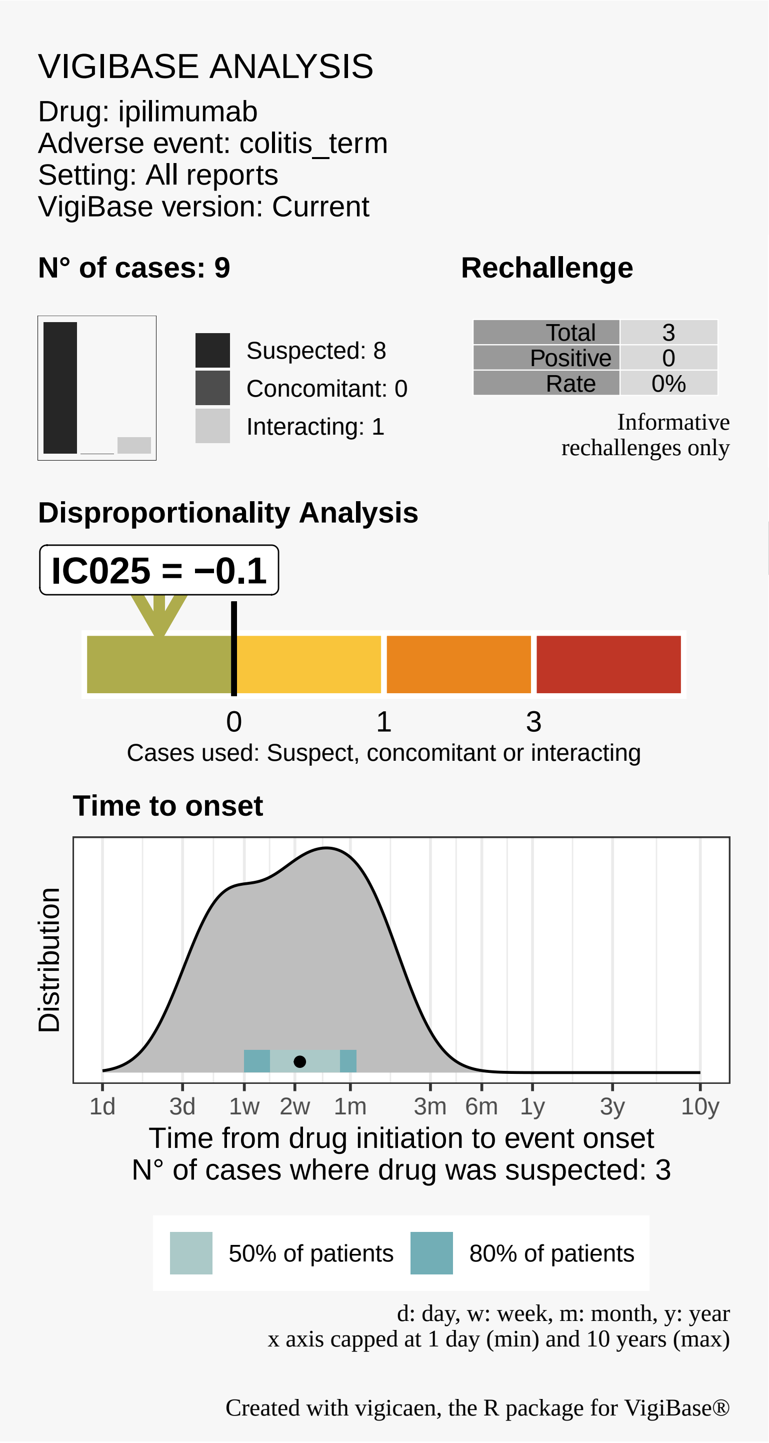
Disproportionality estimates

Although the aim of the package is to prepare readily available datasets for users to compute disproportionality on their own via advanced modelling techniques, it also provides basic estimates through the `compute_dispro()` and `compute_interaction()` functions. The underlying computations rely on the Norén et al methodology, for both point estimates, confidence and credibility intervals. (Norén et al., 2013)

Routine use

As a routine pharmacovigilance practitioner, key information on a drug - adverse event pair may be needed out-of-the-box, without further need for manipulating the underlying tables. To adress the typical needs (disproportionality estimand, time to onset, dechallenge and rechallenge outcomes), `vigi_routine()` creates a graphical output for a given pair. It is intended as a daily practice tool, to support routine assessment of causality. The graph can easily be exported to an external file with the `export_to` argument.

```
vigi_routine(
  demo,
  drug,
  adr_,
  link_,
  d_code = d_drecno,
  a_code = a_llt[1],
  vigibase_version = "Current"
)
```



Conclusion

Easier, reproducible research in pharmacovigilance databases is key to appropriate safety signal detection. Vigicaen proposes a set of tools based on popular open standards to facilitate pharmacovigilance analysis in R.

Acknowledgements

The information presented in this study does not represent the opinion of the Uppsala Monitoring Centre or the World Health Organization. We thank the research team at the Uppsala Monitoring Centre (Uppsala, Sweden) who provided case-level data from VigiBase®.

References

22 arrow – r for data science (2e). (n.d.). https://r4ds.hadley.nz/arrow.html

Alexandre, J., Salem, J.-E., Moslehi, J., Sassier, M., Ropert, C., Cautela, J., Thuny, F., Ederhy, S., Cohen, A., Damaj, G., Vilque, J.-P., Plane, A.-F., Legallois, D., Champ-Rigot, L., Milliez, P., Funck-Brentano, C., & Dolladille, C. (2021). Identification of anticancer drugs associated with atrial fibrillation: analysis of the WHO pharmacovigilance database. *European Heart Journal. Cardiovascular Pharmacotherapy*, *7*(4), 312–320. https://doi.org/10.1093/ehjcvp/pvaa037

Apache arrow. (n.d.). https://arrow.apache.org/

Centre, U. M. (n.d.). *About VigiBase*. https://who-umc.org/vigibase/

Chretien, B., Dolladille, C., Nishida, K., Aleksic, B., Alexandre, J., & L'orphelin, J.-M. (2025). Analysis of anticancer drug associated adverse reactions in depressive patients from vigibase. *Scientific Reports*, *15*(1), 45751. https://doi.org/10.1038/s41598-025-28563-9

Dolladille, C., & Chrétien, B. (2025). *vigicaen: 'VigiBase' Pharmacovigilance Database Toolbox*. https://github.com/pharmacologie-caen/vigicaen

Dolladille, C., Ederhy, S., Sassier, M., Cautela, J., Thuny, F., Cohen, A. A., Fedrizzi, S., Chrétien, B., Da-Silva, A., Plane, A.-F., Legallois, D., Milliez, P. U., Lelong-Boulouard, V., & Alexandre, J. (2020). Immune Checkpoint Inhibitor Rechallenge After Immune-Related Adverse Events in Patients With Cancer. *JAMA Oncology*. https://doi.org/10.1001/jamaoncol.2020.0726

Legallois, D., Da Silva, A., Alexandre, J., Milliez, P., Sabatier, R., Blanchart, K., Plane, A.-F., Font, J., Chrétien, B., & Dolladille, C. (2025). Identification of anticancer drugs associated to cancer therapy-related cardiac dysfunction: a VigiBase® disproportionality analysis. *European Heart Journal. Cardiovascular Pharmacotherapy*, *11*(5), 459–468. https://doi.org/10.1093/ehjcvp/pvaf027

Minoc, E.-M., Villain, C., Chrétien, B., Benbrika, S., Heraudeau, M., Lafont, C., Béchade, C., Lobbedez, T., Lelong-Boulouard, V., & Dolladille, C. (2025). Association between antidepressant drugs and falls in older adults: A mediation analysis in the World Health Organization's pharmacovigilance database. *Therapie*. https://doi.org/10.1016/j.therap.2025.01.004

Montastruc, J.-L., Sommet, A., Bagheri, H., & Lapeyre-Mestre, M. (2011). Benefits and strengths of the disproportionality analysis for identification of adverse drug reactions in a pharmacovigilance database. *British Journal of Clinical Pharmacology*, *72*(6), 905–908. https://doi.org/10.1111/j.1365-2125.2011.04037.x

Nishida, K., Chrétien, B., Dolladille, C., Ebina, T., Aleksic, B., Cabé, N., Savey, V., Onoue, T., & Yatsuya, H. (2025). Psychiatric and psychological adverse effects associated with dulaglutide, semaglutide, and liraglutide: A vigibase study. *Clinical Nutrition*, *51*, 252–265. https://doi.org/10.1016/j.clnu.2025.06.011

Norén, G. N., Hopstadius, J., & Bate, A. (2013). Shrinkage observed-to-expected ratios for robust and transparent large-scale pattern discovery. *Statistical Methods in Medical Research*, *22*(1), 57–69. https://doi.org/10.1177/0962280211403604

Parquet. (n.d.). https://parquet.apache.org/

R packages (2e). (n.d.). https://r-pkgs.org/

Richardson, N., Cook, I., Crane, N., Dunnington, D., François, R., Keane, J., Moldovan-Grünfeld, D., Ooms, J., Wujciak-Jens, J., Luraschi, J., Werner, K. D., Wong, J., & Arrow, A. (2025). *Arrow: Integration to 'apache' 'arrow'*. https://cran.r-project.org/web/packages/arrow/index.html

Wickham, H., & RStudio. (2023). *Tidyverse: Easily install and load the 'tidyverse'*. https://cran.r-project.org/web/packages/tidyverse/index.html